THE INDIVIDUAL STATION TECHNIQUE FOR THE ANALYSIS OF CONTINUOUS-TIME POLLING SYSTEMS

Mandyam M. Srinivasan

Department of
Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 91-36

December 1991

The Individual Station Technique for the Analysis of Continuous-Time Polling Systems

Mandyam M. Srinivasan

Department of
Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 91-36
December 1991

Abstract

Polling systems find application in a wide variety of real-world systems, for example, in telecommunication systems and material handling systems, and there is continued interest in the analysis of polling systems for their performance. In this paper, we present a very promising technique to obtain mean waiting times for continuous-time polling systems which use the exhaustive and gated service disciplines. The technique enables the analysis of polling systems which have a mix of these service disciplines. Thus, some nodes could adopt the exhaustive service discipline while the other nodes use the gated service discipline. The technique also allows the user to obtain the mean waiting time at select nodes without having to obtain all the mean waiting times simultaneously.

A polling system is a queueing system in which a single server services M nodes. The server cycles around the system, visiting each node in turn, and attends to the waiting customers using some prespecified service discipline. Real-world applications which are modeled and analyzed using polling systems can be found, for example, in telecommunication and material handling systems, and there is continued interest in developing new algorithms and applications for polling systems. An excellent survey of the research done on polling systems can be found in Takagi (1990).

A number of service disciplines have been studied, which determine the service strategy adopted by the server at a node. The most commonly studied service disciplines are i) the exhaustive service discipline, where the server continues to serve all customers at the node until there are no more customers at the node, ii) the gated service discipline, where the server only serves those customers present at the node at the instant the node is polled, and iii) the k-limited service discipline in which the server serves at most k customers during a visit to that node. Several variants of these strategies have been proposed and studied in the past.

In this paper we present a new and powerful technique for the analysis of continuous-time polling systems, which we term the Individual Station (IS) technique. The IS technique allows us to determine the mean waiting time at one or more select nodes without having to obtain mean waiting times at all the nodes, simultaneously. We develop the following new results: a) We provide an iterative algorithm to determine the exact mean waiting times for the asymmetric, continuous-time, exhaustive and gated service polling systems. b) We next present an algorithm which computes mean waiting times very rapidly without recourse to iteration. The mean waiting time at a node is determined, *in an explicit form* (through the solution to a system of six equations or less), and requires $O(M^2)$ operations per node. The algorithm is numerically exact for polling systems with upto seven nodes. We demonstrate that for polling systems with an arbitrarily larger number of nodes, the solution of this system of equations results in mean waiting time estimates very close to the exact values, with errors *less than 0.0025%*, even when utilizations approach 1. For the exhaustive service polling system with more than seven nodes, we also show that the mean waiting time obtained from the algorithm bounds the exact numerical value from below. If mean waiting times need to be computed even more accurately, they can be obtained with very little additional effort, simply by solving a slightly larger system of equations (say, a system of seven equations). c) We obtain the mean waiting times for a polling system which allows mixed service strategies, wherein some of the nodes are served using the exhaustive service discipline, while the other nodes are served using the gated service discipline.

1. Literature Review

There is a considerable amount of literature devoted to polling systems analysis, and the reader is referred to the paper by Takagi (1990) for a comprehensive survey of the work done in this area. We will briefly review some of the papers most pertinent to our work.

A number of numerical techniques have been proposed for computing the mean waiting times for the exhaustive and gated polling systems. Levy (1991) classifies these techniques as

a) The buffer-occupancy equation method (Cooper and Murray 1969, Cooper and Murray 1970, Eisenberg 1972, Hashida 1972, Konheim and Meister 1974, Rubin and Demoraes1983, Kleinrock and Levy 1988, and Takagi 1986),
b) The station-time equation method (Aminetzah 1975, Humblet 1978, Ferguson and Aminetzah 1985, and Baker and Rubin 1987),
c) Carsten, Newhall and Posner's method (1977),
d) Swartz's method (1980), and
e) Sarkar and Zangwill's method (1989).

Most of the above methods require the solution of large sets of linear equations (either M^2 or M^3 equations). The algorithm of Sarkar and Zangwill, however, obtains the mean waiting times by solving a system of M linear equations, thereby requiring only $O(M^3)$ operations.

Swartz (1980) develops an iterative procedure for analyzing the discrete-time exhaustive service system, in which time is slotted in fixed intervals. Although this procedure does not guarantee an upper bound on the number of iterations required, it converges rather quickly when the utilization of the server is not very close to 1. Levy (1989, 1991) discusses the complexity of this algorithm, and the rate at which it converges. Levy (1991) observes that Swartz's method can be used to calculate the mean waiting time at a station independently of the mean waiting times at other stations. This is in contrast to the other methods which solve for all M mean waiting times, simultaneously. Levy shows that the method forms a contraction mapping and, therefore, the number of iterations it requires is logarithmic in the accuracy required. Levy concludes that "for a wide range of parameters, the method is the most efficient method known today for computing the expected delay in polling systems," and that it is desirable to apply this approach to polling systems other than the discrete-time exhaustive service system.

The iterative algorithm we present in this paper (for the continuous-time exhaustive and gated service polling systems) is similar to Swartz's approach for the discrete-time exhaustive service system. It can, therefore, be shown to share the above property of logarithmic convergence.

We first consider the exhaustive service system, and derive the expression for the mean waiting time in Section 2. In Section 3, we present the iterative algorithm, using the IS technique. We next develop the algorithm which requires $O(M^2)$ operations to obtain the mean waiting time at a node. Section 4 presents a number of computational results for the two algorithms. These results serve to illustrate both the speed of convergence of the iterative algorithm, and demonstrate the accuracy of the second algorithm. In section 5, we extend the results of sections 3 and 4 to the gated service system. In section 6, we show how the IS technique easily extends to handle polling systems in which some of the nodes adopt the exhaustive service discipline while the others adopt the gated service discipline. Section 7 presents a summary and discusses ways in which the IS technique could be further extended.

2. The Model for the Exhaustive Service Polling System

We present most of the notation that will be used in the paper. Unless otherwise specified, it is implicit that the index for any summation is over the range 1 through M. It is also implicit that the index is i) reset to 1 if it becomes M+1, and ii) reset to M if it drops down to 0. We adopt the convention that an empty product equals 1.

A single server serves requests from customers at the M nodes, according to the exhaustive service discipline. Customers arrive at each node according to independent Poisson processes with rate λ_m at node m. The service time of a customer at node m is an independent random variable, having finite mean b_m and second moment $b^{(2)}_m$. The Laplace Stieltjes Transform (LST) of the service time is denoted by $\beta_m(.)$, and the LST of the busy period (Cooper 1981) induced by a single customer at node m is denoted by $\eta_m(.)$. The time taken by the server to switch between nodes m and m+1 is a random variable with mean s_m, second moment $s^{(2)}_m$, variance $Var(S_m)$, and LST $\sigma_m(.)$. The traffic intensity at node m is denoted by $\rho_m = \lambda_m b_m$, and $\rho = \sum_m \rho_m$ denotes the server utilization. For the polling system to be stable, we require ρ to be less than one. The sum of the mean switchover times is denoted by $s = \sum_m s_m$. We are interested in obtaining the mean waiting time, W_m, at node m, for m=1,...,M.

Let $\mathbf{z} = (z_1,\ldots,z_M)$, and let $F_m(\mathbf{z})$ denote the probability generating function for the number of customers present at each node at the instant the server polls node m. The first and second factorial moments of the number of customers present at node m when the server polls the node are given by f_m and $f^{(2)}_m$, respectively, where

$$f_m = \frac{\partial F_m(\mathbf{z})}{\partial z_m}\Big|_{\mathbf{z}=\mathbf{1}}, \qquad \text{and} \qquad f^{(2)}_m = \frac{\partial^2 F_m(\mathbf{z})}{\partial {z_m}^2}\Big|_{\mathbf{z}=\mathbf{1}}. \tag{2.1}$$

If we can obtain the first and second factorial moments of the number of customers present at node m when the server polls that node, then the mean waiting time is obtained as (Takagi 1986):

$$W_m = \frac{f_m^{(2)}}{2\lambda_m f_m} + \frac{\lambda_m b_m^{(2)}}{2(1-\rho_m)}. \tag{2.2}$$

Obtaining these moments is not straightforward, since the generating function $F_m(.)$ is expressed, recursively, in terms of the generating function $F_{m-1}(.)$. It is well known (see for instance, Takagi 1986) that the generating functions are related by the following expression:

$$F_{m+1}(z_1,\ldots,z_M) = F_m(z_1,\ldots,z_{m-1},\eta_m(\Sigma k{\neq}m\ (\lambda_k-\lambda_k z_k)),z_{m+1},\ldots,z_M)\ \sigma_m(\Sigma k\ (\lambda_k-\lambda_k z_k)). \tag{2.3}$$

This equation forms the starting point for the IS technique. We initially obtain an expression which expresses $F_m(.)$ solely in terms of functions of input parameters. We then differentiate the resulting expression twice and use equation (2.2) to obtain the mean waiting times. The technique we develop below, computes W_1. The mean waiting times at the other nodes are obtained in an identical manner. We first define a recursive (nested) function, $\gamma_m(j)$ as follows:

$$\gamma_m(0) = z_m, \qquad m=1,\ldots,M, \tag{2.4a}$$

$$\gamma_m(j) = \eta_m\Big(\sum_{k<m} [\lambda_k-\lambda_k\gamma_k(j-1)] + \sum_{k>m} [\lambda_k-\lambda_k\gamma_k(j)]\Big), \quad j > 0,\ m = 1,\ldots,M. \tag{2.4b}$$

Let

$$\sigma_m(j) = \sigma_m\Big(\sum_{k\leq m} [\lambda_k-\lambda_k\gamma_k(j)] + \sum_{k>m} [\lambda_k-\lambda_k\gamma_k(j+1)]\Big), \quad j \geq 0,\ m = 1,\ldots,M. \tag{2.5}$$

We can now cast equation (2.3) in the following form:

$$F_1(\gamma_1(0),\ldots,\gamma_M(0)) = F_M(\gamma_1(0),\ldots,\gamma_{M-1}(0),\gamma_M(1))\ \sigma_M(0).$$

Recursively expressing $F_M(.)$ in terms of $F_{M-1}(.)$, and $F_{M-1}(.)$ in terms of $F_{M-2}(.)$, and so on, we obtain:

$$F_1(\gamma_1(0),\ldots,\gamma_M(0)) = F_1(\gamma_1(1),\ldots,\gamma_M(1)) \prod_m \sigma_m(0).$$

We now continue to recursively express $F_1(\gamma_1(j),\ldots,\gamma_M(j))$ in terms of $F_1(\gamma_1(j+1),\ldots,\gamma_M(j+1))$, for $j \geq 1$, and this results in the following expression for $F_1(\gamma_1(0),\ldots,\gamma_M(0))$:

$$F_1(\gamma_1(0),\ldots,\gamma_M(0)) = F_1(\gamma_1(n),\ldots,\gamma_M(n)) \prod_{j=0}^{n-1} \Big(\prod_m \sigma_m(j)\Big). \tag{2.6}$$

It can be shown (refer Cooper 1969, and Eisenberg 1972, for example), that $\lim_{n\to\infty}\gamma_1(n) = 1$. Hence, letting $n \to \infty$ in equation (2.6), we obtain:

$$F_1(\gamma_1(0),\ldots,\gamma_M(0)) \quad = \quad \prod_{j=0}^{\infty} \left(\prod_{m} \sigma_m(j)\right) \tag{2.7}$$

We now have an equation for F_1 that is only in terms of (complex) functions of the $\sigma_m(j)$ values. Differentiating equation (2.7) once and twice with respect to z_1, and setting $\mathbf{z} = \mathbf{1}$, we obtain:

$$f_1 \quad = \quad \sum_{j=0}^{\infty} \sum_{m} \frac{\partial \sigma_m(j)}{\partial z_1}\Big|_{\mathbf{z}=\mathbf{1}}, \qquad \text{and} \tag{2.8a}$$

$$f_1^{(2)} \quad = \quad f_1^2 + \sum_{j=0}^{\infty} \sum_{m} \frac{\partial^2 \sigma_m(j)}{\partial z_1^2}\Big|_{\mathbf{z}=\mathbf{1}} - \sum_{j=0}^{\infty} \sum_{m} \left(\frac{\partial \sigma_m(j)}{\partial z_1}\right)^2\Big|_{\mathbf{z}=\mathbf{1}}. \tag{2.8b}$$

Let

$$\psi_m(j) \; = \; \lambda_m \frac{\partial \gamma_m(j)}{\partial z_1}\Big|_{\mathbf{z}=\mathbf{1}}, \qquad \psi_m^{(2)}(j) \; = \; \lambda_m \frac{\partial^2 \gamma_m(j)}{\partial z_1^2}\Big|_{\mathbf{z}=\mathbf{1}}, \quad \text{and} \tag{2.9}$$

$$\delta_m \quad = \quad \frac{\rho_m}{1-\rho_m} \tag{2.10}$$

We obtain $\psi_m(j)$ and $\psi_m^{(2)}(j)$, for $j > 0$, by differentiating equation (2.4b) once and twice with respect to z_1, and then setting $\mathbf{z}$ equal to $\mathbf{1}$, to get:

$$\psi_m(j) = \quad \delta_m \Big[\sum_{k<m} \psi_k(j-1) \; + \; \sum_{k>m} \psi_k(j)\Big], \quad j > 0, \text{ and} \tag{2.11a}$$

$$\psi_m^{(2)}(j) \; = \; \delta_m \Big[\sum_{k<m} \psi_k^{(2)}(j-1) + \sum_{k>m} \psi_k^{(2)}(j)\Big] + \frac{\lambda_m b_m^{(2)}}{(1-\rho_m)^3} \Big[\sum_{k<m} \psi_k(j-1) + \sum_{k>m} \psi_k(j)\Big]^2, \; j > 0. \tag{2.11b}$$

In a similar manner, from equation (2.5) we obtain, for $j \geq 0$:

$$\frac{\partial \sigma_m(j)}{\partial z_1}\Big|_{\mathbf{z}=\mathbf{1}} \; = \; s_m \Big[\sum_{k\leq m} \psi_k(j) \; + \; \sum_{k>m} \psi_k(j+1)\Big], \qquad \text{and} \tag{2.12a}$$

$$\frac{\partial^2 \sigma_m(j)}{\partial z_1^2}\Big|_{\mathbf{z}=\mathbf{1}} \; = \; s_m \Big[\sum_{k\leq m} \psi_k^{(2)}(j) + \sum_{k>m} \psi_k^{(2)}(j+1)\Big] \; + \; s_m^{(2)} \Big[\sum_{k\leq m} \psi_k(j) + \sum_{k>m} \psi_k(j+1)\Big]^2. \tag{2.12b}$$

Hence, from equations (2.8a) and (2.12a), interchanging the order of summations, we obtain

$$f_1 \quad = \quad \sum_{m} s_m \sum_{j=0}^{\infty} \Big[\sum_{k\leq m} \psi_k(j) \; + \; \sum_{k>m} \psi_k(j+1)\Big]. \tag{2.13}$$

Observation 1: a) The term $\psi_k(0) = \lambda_k \frac{\partial z_k}{z_1}\Big|_{\mathbf{z}=\mathbf{1}} = \lambda_1$ when k=1; it is equal to zero otherwise. b) The term $\psi_k^{(2)}(0) = 0$ for all k. ■

Based on Observation 1a), we can rewrite equation (2.13) as

$$f_1 \;=\; \sum_m s_m \sum_{j=0}^{\infty} \sum_k \psi_k(j) \;=\; s \sum_{j=0}^{\infty} \sum_k \psi_k(j) \;=\; s \sum_k \sum_{j=0}^{\infty} \psi_k(j). \tag{2.14}$$

Let

$$x_k \;=\; \textstyle\sum_{j=0}^{\infty} \psi_k(j). \tag{2.15}$$

To evaluate x_k, we use Observation 1a) once more. From equation (2.11a), we obtain:

$$x_m \;=\; \begin{cases} \lambda_1 + \delta_1 \sum_{k\neq 1} x_k, & m = 1, \\ \delta_m \sum_{k\neq m} x_k, & \text{otherwise.} \end{cases}$$

We add $\delta_m x_m$ to both sides of the above equation, and divide both sides of the resulting identity by $(1+\delta_m)$. Next, we use the following facts: a) $\delta_m/(1+\delta_m) = \rho_m$, and b) $1/(1+\delta_m) = (1-\rho_m)$. This results in the following expression for x_m:

$$x_m \;=\; \begin{cases} (1-\rho_1)\lambda_1 + \rho_1 \sum_k x_k, & m = 1, \\ \rho_m \sum_k x_k, & \text{otherwise.} \end{cases}$$

We now sum x_m over all m, to get:

$$\textstyle\sum_k x_k \;=\; \lambda_1 \frac{1-\rho_1}{1-\rho}. \tag{2.16}$$

From equations (2.15) and (2.16), we thus obtain:

$$f_1 \;=\; \lambda_1\, s \frac{1-\rho_1}{1-\rho}. \tag{2.17}$$

We can now obtain $f_1^{(2)}$ using an approach similar to that used to obtain f_1. From equations (2.8b), (2.12a), and (2.12b), we obtain the following expression for $f_1^{(2)}$:

$$f_1^{(2)} \;=\; f_1^2 + \sum_{j=0}^{\infty} \sum_m \Big(s_m \big[\sum_{k\leq m} \psi_k^{(2)}(j) + \sum_{k>m} \psi_k^{(2)}(j+1) \big] + \mathrm{Var}(S_m) \big[\sum_{k\leq m} \psi_k(j) + \sum_{k>m} \psi_k(j+1) \big]^2 \Big).$$

Let $y_k = \sum_{j=1}^{\infty} \psi_k^{(2)}(j)$. Interchanging the order of summation wherever possible in the above equation, and using Observation 1b) to note that $\psi_k^{(2)}(0) = 0$ for all k, we get:

$$f_1^{(2)} \;=\; f_1^2 + \sum_m s_m \sum_k y_k + \sum_m \mathrm{Var}(S_m) \sum_{j=0}^{\infty} \big[\sum_{k\leq m} \psi_k(j) + \sum_{k>m} \psi_k(j+1) \big]^2. \tag{2.18}$$

Similar to the approach used to determine $\sum_m x_m$, we obtain $\sum_m y_m$, after some algebra, as

$$\sum_m y_m \;=\; \sum_m \frac{\lambda_m b_m^{(2)}}{(1-\rho)\rho_m^2} \sum_{j=1}^{\infty} \psi_m^2(j). \tag{2.19}$$

From equation (2.11a), for m=1,...,M−1,

$$[\sum_{k\le m} \psi_k(j) + \sum_{k>m} \psi_k(j+1)]^2 = [\sum_{k\le m} \psi_k(j) + \psi_{m+1}(j+1) + \sum_{k>m+1} \psi_k(j+1)\,]^2$$

$$= [\frac{\psi_{m+1}(j+1)}{\delta_{m+1}} + \psi_{m+1}(j+1)]^2 = \frac{\psi_{m+1}^2(j+1)}{\rho_{m+1}^2}. \tag{2.20}$$

For m=M, from equation (2.11a):

$$[\sum_{k\le M} \psi_k(j)]^2 = \lambda_1^2,\ j=0, \quad \text{and} \quad [\sum_{k\le M} \psi_k(j)]^2 = \frac{\psi_1^2(j)}{\rho_1^2},\quad j>0. \tag{2.21}$$

Hence, from equations (2.18) through (2.21), we obtain:

$$f_1^{(2)} = f_1^2 + s\sum_m \frac{\lambda_m b_m^{(2)}}{(1-\rho)\rho_m^2} \sum_{j=1}^{\infty} \psi_m^2(j) + \sum_m \frac{\mathrm{Var}(S_m)}{\rho_m^2} \sum_{j=1}^{\infty} \psi_{m+1}^2(j) + \lambda_1^2\,\mathrm{Var}(S_M). \tag{2.22}$$

We can rewrite equation (2.22) as

$$f_1^{(2)} = f_1^2 + s\sum_m \frac{\lambda_m b_m^{(2)}}{(1-\rho)\rho_m^2} \sum_{j=1}^{\infty} \psi_m^2(j) + \sum_m \frac{\mathrm{Var}(S_{m-1})}{\rho_m^2} \sum_{j=1}^{\infty} \psi_m^2(j) + \lambda_1^2\,\mathrm{Var}(S_M). \tag{2.23}$$

It is now convenient to define two terms, $\chi_m(j)$ and $\varphi_m(n)$, as follows.

$$\chi_m(j) = \frac{\psi_m(j)}{\lambda_1} = \frac{\lambda_m}{\lambda_1}\frac{\partial \gamma_m(j)}{\partial z_1}, \quad \text{and} \tag{2.24a}$$

$$\varphi_m(n) = \sum_{j=n+1}^{\infty} \chi_m(j)\chi_m(j-n). \tag{2.24b}$$

From equations (2.23) and (2.24), we get an expression for $f_1^{(2)}$ given below as Lemma 2.1.

Lemma 2.1:

$$f_1^{(2)} = f_1^2 + \lambda_1^2 \sum_m \varphi_m(0)\,(\frac{s}{1-\rho}\frac{\lambda_m b_m^{(2)}}{\rho_m{}^2} + \frac{\mathrm{Var}(S_{m-1})}{\rho_m{}^2}) + \lambda_1^2\,\mathrm{Var}(S_M). \qquad \blacksquare$$

From Lemma 2.1 and equation (2.17), we obtain the main result of this section, stated as:

Theorem 2.2: The mean waiting time at node 1 for the exhaustive service system is:

$$W_1 = \frac{s}{2}\frac{1-\rho_1}{1-\rho} + \sum_m \frac{\varphi_m(0)}{(1-\rho_1)\rho_m^2}(\frac{\lambda_m b_m^{(2)}}{2} + \frac{1-\rho}{2s}\mathrm{Var}(S_{m-1})) + (\frac{\lambda_1 b_1^{(2)}}{2(1-\rho_1)} + \frac{1-\rho}{2s(1-\rho_1)}\mathrm{Var}(S_M)). \tag{2.25}$$

$\blacksquare$

The expressions for the mean waiting times at the other nodes can be obtained in a similar manner, simply by renumbering the indices. Thus, obtaining the mean waiting times reduces to computing the term $\varphi_m(0)$, since the other terms are just input data. In the following section, we develop the two algorithms to compute $\varphi_m(0)$.

3. Computing $\varphi_m(0)$ for the exhaustive service system

The two algorithms we present in this section differ in computing $\varphi_m(0)$ as follows. The first algorithm iteratively computes $\chi_m^2(j)$, starting with $j = 1$ until the terms approach 0. In the second algorithm, we only compute $\chi_m^2(j)$ for $j=1,\ldots,M-1$, and then solve a system of equations as explained subsequently.

We first present a simple expression for recursively computing $\chi_m(j)$. From equations (2.11a) and (2.24a), after some elementary algebra, we obtain:

$$\chi_M(1) = \delta_M; \qquad \chi_m(1) = (\chi_{m+1}(1)/\rho_{m+1})\delta_m, \quad m=1,\ldots,M-1, \tag{3.1a}$$

and for $j > 1$,

$$\chi_M(j) = (\chi_1(j-1)/\rho_1 - \chi_M(j-1))\delta_M; \quad \chi_m(j) = (\chi_{m+1}(j)/\rho_{m+1} - \chi_m(j-1))\delta_m, \quad m=1,\ldots,M-1. \tag{3.1b}$$

Algorithm 1:

0. Set $j = 1$, and compute $\chi_m(j)$, $m=1,\ldots,M$, using equation (3.1a). Initialize $\varphi_m(0) = \chi_m^2(j)$ for $m = 1,\ldots,M$. Choose a tolerance value ε.
1. Set $j = j+1$.
2. Compute $\chi_m(j)$, using equation (3.1b), and set $\varphi_m(0) = \varphi_m(0) + \chi_m^2(j)$, $m = 1,\ldots,M$. If $\chi_m(j) \leq \varepsilon$ for all m, stop. Otherwise go to step 1.
3. Obtain the mean waiting time at node 1, using equation (2.25).

Algorithm 1 is executed for each node whose mean waiting time is desired. Since computation of $\varphi_m(0)$ only needs the terms ρ_m and $\delta_m = \rho_m/(1-\rho_m)$, we can compute the mean waiting time for, say, node 2 simply by using a temporary variable, $\tilde{\rho}_m$, to store permuted ρ_m values. To compute W_2, we just set $\tilde{\rho}_m = \rho_{m+1}$, $m=1,\ldots,M$, and (re)compute new δ_m values using the $\tilde{\rho}_m$ values.

For Algorithm 2, we need to define some additional terms, Γ_m, $m=1,\ldots,M$. Let

$$\Gamma_M = \prod_m \delta_m, \qquad \text{and} \qquad \Gamma_m = \Gamma_{m+1} + \sum_{k=m}^{M} \kappa_m^{(i)}, \quad 1 \leq m < M, \quad \text{where} \tag{3.2}$$

$$\kappa_m^{(1)} = \delta_m, \qquad \text{and} \qquad \kappa_m^{(i)} = \delta_m \sum_{j=i-1}^{m-1} \kappa_j^{(i-1)}, \quad m = 1,\ldots,M. \tag{3.3}$$

Intuitively, $\kappa_m^{(i)}$ denotes the sum of all possible combinations of i unique δ_k terms, such that δ_m is in every one of these combinations. The term Γ_m denotes the sum of all possible combinations of m unique δ_k terms, plus the sum of all possible combinations of m+1 unique δ_k terms + ... + the sum of all possible combinations of M unique δ_k terms. (There is, of course, only one combination of M unique δ_k terms.) Note that all the Γ_m terms, m=1,...,M, are strictly positive. Lemma 3.1 shows that the sum of Γ_m, m = 2,...,M is less than one. The proof of Lemma 3.1 easily follows by induction on M, and it is omitted here.

Lemma 3.1:

$$(1-\sum_{n=2}^{M}\Gamma_n)\prod_{m}(1-\rho_m) \quad = \quad 1-\rho. \qquad \blacksquare$$

The next lemma, Lemma 3.2, expresses $\chi_m(j)$ in terms of $\chi_m(j-n)$, for n>0. The proof of Lemma 3.2 follows by induction on the number of nodes and is omitted. Although Lemma 3.2 requires the definition of an additional variable, $\theta_m^{(i)}$, we remark that this variable is needed only to establish some initial conditions. It is not used subsequently. The variables $\theta_m^{(i)}$ are defined recursively as follows:

$$\theta_m^{(1)} = \delta_m, \qquad \text{and for } i > 1, \quad \theta_m^{(i)} = \kappa_m^{(i)} - \delta_1\,\theta_m^{(i-1)}.$$

Lemma 3.2:

$$\chi_m(j) = \sum_{n=1}^{M-1}\chi_m(j-n)\,\Gamma_{n+1}, \quad j \geq M, \; m = 1,\ldots,M, \quad \text{with} \tag{3.4}$$

$$\chi_m(j) = \sum_{n=1}^{j}\chi_m(j-n)\,\Gamma_{n+1} + \theta_m^{(j)}\prod_{n=m+1}^{M}(1+\delta_n), \quad j < M, \; m = 2,\ldots,M, \tag{3.5a}$$

$$\chi_1(j) = \sum_{n=1}^{j}\chi_1(j-n)\,\Gamma_{n+1} - \sum_{m=j+2}^{M}\theta_m^{(j+1)}\prod_{n=m+1}^{M}(1+\delta_n), \quad j < M. \tag{3.5b}$$

$$\blacksquare$$

We now show how Lemma 3.2 enables us to obtain the mean waiting time at a node. To clarify the technique that we subsequently develop for the general case, we first consider the special case for M=3 nodes. It will be convenient to define:

$$\tilde{\varphi}_m(n) = \sum_{j=M}^{\infty}\chi_m(j)\chi_m(j-n). \tag{3.6}$$

Note that

$$\varphi_m(n) = \tilde{\varphi}_m(n) + \sum_{j=1}^{M-1} \chi_m(j)\chi_m(j-n). \tag{3.7}$$

We remind the reader that we are interested in computing $\varphi_m(0)$.

3.1. Algorithm 2: The special case of M=3 nodes

For this case, we have from Lemma 3.2,

$$\tilde{\varphi}_m(0) = \sum_{j=3}^{\infty} \chi_m^2(j) = \sum_{j=3}^{\infty} [\Gamma_2\,\chi_m(j-1) + \Gamma_3\,\chi_m(j-2)]^2$$

$$= \sum_{j=3}^{\infty} [\Gamma_2^2\,\chi_m^2(j-1) + \Gamma_3^2\,\chi_m^2(j-2) + 2\Gamma_2\Gamma_3\chi_m(j-1)\chi_m(j-2)]$$

$$= \Gamma_2^2\chi_m^2(2) + \Gamma_2^2\,\tilde{\varphi}_m(0) + \Gamma_3^2\,(\chi_m^2(1)+\chi_m^2(2)) + \Gamma_3^2\,\tilde{\varphi}_m(0) + 2\Gamma_2\Gamma_3\chi_m(1)\chi_m(2) + 2\Gamma_2\Gamma_3\tilde{\varphi}_m(1).$$

The term $\tilde{\varphi}_m(1)$ is, in turn, obtained from another application of Lemma 3.2 as:

$$\tilde{\varphi}_m(1) = \sum_{j=3}^{\infty} \chi_m(j)\chi_m(j-1) = \sum_{j=3}^{\infty} [\Gamma_2\,\chi_m(j-1) + \Gamma_3\,\chi_m(j-2)]\,\chi_m(j-1)$$

$$= \Gamma_2\,\chi_m^2(2) + \Gamma_2\,\tilde{\varphi}_m(0) + \Gamma_3\chi_m(1)\chi_m(2) + \Gamma_3\,\tilde{\varphi}_m(1)$$

This results in two equations in the two unknowns, $\tilde{\varphi}_m(0)$ and $\tilde{\varphi}_m(1)$. Solving for $\tilde{\varphi}_m(0)$, we get:

$$\tilde{\varphi}_m(0) = \frac{\chi_m^2(2)\,(\Gamma_2^2 + \Gamma_3^2 + 2\Gamma_2^2\,\Gamma_3/(1-\Gamma_3)) + \chi_m^2(1)\,\Gamma_3^2 + 2\chi_m(1)\,\chi_m(2)\,\Gamma_2\,\Gamma_3/(1-\Gamma_3)}{1-\Gamma_2^2 - \Gamma_3^2 - 2\Gamma_2^2\,\Gamma_3/(1-\Gamma_3)}. \tag{3.8a}$$

To compute $\varphi_m(0)$, from equation (3.7) we add $\chi_m^2(1)+\chi_m^2(2)$ to $\tilde{\varphi}_m(0)$, and simplify the result to get

$$\varphi_m(0) = \chi_m^2(1) + \frac{\chi_m^2(2) + \chi_m^2(1)\,\Gamma_3^2 + 2\chi_m(1)\,\chi_m(2)\,\Gamma_2\,\Gamma_3/(1-\Gamma_3)}{1 - \Gamma_2^2 - \Gamma_3^2 - 2\Gamma_2^2\,\Gamma_3/(1-\Gamma_3)}. \tag{3.8b}$$

Note that for three nodes, Γ_2 and Γ_3 are computed very simply as $\Gamma_3 = \delta_1\delta_2\delta_3$, and $\Gamma_2 = \Gamma_3 + \delta_1\delta_2 + \delta_2\delta_3 + \delta_3\delta_1$. Note, too, that we only compute Γ_2 and Γ_3 once, since these terms are permutation-invariant. However, for each node for which mean waiting times are desired, we must compute the $\chi_m(1)$ and $\chi_m(2)$ terms, m=1,...M, after permuting the node indices as mentioned earlier. We can directly substitute the expression for $\varphi_m(0)$, obtained from equation (3.8b), in equation (2.25) to obtain the mean waiting time.

3.2. Algorithm 2: The general case

For an arbitrary number of nodes, using a similar approach as in section 3.1, we can show that $\varphi_m(0)$ is obtained by solving the following system of equations:

$$\tilde{\varphi}_m(0) = b_m(0) + \tilde{\varphi}_m(0) \sum_{n=2}^{M} \Gamma_n^2 + 2 \sum_{i=1}^{M-2} \tilde{\varphi}_m(i) \sum_{n=2}^{M-i} \Gamma_n \Gamma_{n+i}, \tag{3.9a}$$

$$\tilde{\varphi}_m(k) = b_m(k) + \sum_{n=1-k}^{M-k-1} \tilde{\varphi}_m(n)\Gamma_{n+k+1}, \qquad M-2 \geq k > 0. \tag{3.9b}$$

In equation (3.9b), we adopt the convention that $\tilde{\varphi}_m(-n) = \tilde{\varphi}_m(n)$, $n \geq 0$. In general, to obtain the $\tilde{\varphi}_m(0)$ values for a system with M nodes, we first need to evaluate Γ_n from n=2 upto n=M. The terms $b_m(k)$ in the above equations are data and are obtained as follows:

$$b_m(0) = \sum_{n=1}^{M-1} g_m^2(n), \qquad b_m(k) = \sum_{n=1}^{k} \chi_m(M-1-k+n)\, g_m(n), \quad k = 1,\ldots,M-2, \tag{3.10}$$

where $g_m(n)$ is computed, recursively, from

$$g_m(n) = \sum_{k=n}^{M-1} \chi_m(M-1-k+n)\, \Gamma_{k+1}. \tag{3.11}$$

We remind the reader that in order to compute the mean waiting time at a node, the $b_m(n)$ terms have to be computed for each m. Computing the $b_m(n)$ terms from equations (3.10) and (3.11) would require $O(M^2)$ effort for each m, and hence it would require $O(M^3)$ operations to determine all the coefficients.

However, as we now show, it is not necessary to evaluate all the $b_m(n)$ terms. This is based on the fact that the Γ_n terms converge to 0 very rapidly. For example, suppose we make the reasonable assumption that the value of Γ_n is maximized, for any n>2, when the δ_m values are all the same. Under this assumption, the largest possible for δ_m would result when ρ approaches 1 and, since all ρ_m values are equal to ρ/M, δ_m would approach 1/(M-1). Since Γ_n is the sum of all possible combinations of n unique δ_k terms plus the sum of all possible combinations of n+1 unique δ_k terms + ... + the sum of all possible combinations of M unique δ_k terms, this, in turn, implies that the largest possible value for Γ_n would be $\sum_{i=n}^{M} \binom{M}{i}\left(\frac{1}{M-1}\right)^i$. We know that the sum $\sum_{i=0}^{M} \binom{M}{i}\left(\frac{1}{M-1}\right)^i = \left(1 + \frac{1}{M-1}\right)^{M-1}$, and that $\lim_{M\to\infty}\left(1 + \frac{1}{M-1}\right)^{M-1} = e$. This, in conjunction with Lemma 3.1, leads us to the following observation:

Observation 2. The terms Γ_n decrease at an exponential rate with increasing n. For $n \geq 2$, the magnitude of Γ_n is on the order of $\frac{1}{n!}$. ■

Based on the above observation, instead of solving the system of equations specified by (3.9), we consider solving a reduced system of equations, ignoring terms involving Γ_k, for all k greater than some n. We ran a number of experiments, with values of ρ ranging from 0.3 to 0.996, and with the number of nodes ranging from 3 to 50. In all cases, the value of Γ_8 was found to be *less than* 0.000015. Based on these results, we concluded that it was reasonable to consider solving a reduced system of equations in which all terms involving Γ_n, $n \geq 8$, were disregarded. Thus, we set $L = \min(M,7)$, and content ourselves with solving the following system of equations:

$$\tilde{\varphi}_m(0) = b_m(0) + \tilde{\varphi}_m(0) \sum_{n=2}^{L} \Gamma_n^2 + 2 \sum_{i=1}^{L-2} \tilde{\varphi}_m(i) \sum_{n=2}^{L-i} \Gamma_n \Gamma_{n+i}, \tag{3.12a}$$

$$\tilde{\varphi}_m(k) = b_m(k) + \sum_{n=1-k}^{L-k-1} \tilde{\varphi}_m(n) \Gamma_{n+k+1}, \quad L-2 \geq k > 0, \quad \text{with} \tag{3.12b}$$

$$b_m(0) = \sum_{n=1}^{L-1} g_m^2(n), \qquad b_m(k) = \sum_{n=1}^{k} \chi_m(M-1-k+n)\, g_m(n), \quad k = 1,\ldots,L\text{-}2, \quad \text{and} \tag{3.13}$$

$$g_m(n) = \sum_{k=n}^{L-1} \chi_m(M-1-k+n)\, \Gamma_{k+1}. \tag{3.14}$$

We will refer to Algorithm 2 with $L = \min(M,7)$, as Algorithm 2 at "level 7." Note that Algorithm 2 at level 7 returns the exact numerical solution if the number of nodes is 7 (or less). We can rewrite the system of equations given by (3.12) as $\mathbf{Bx} = \mathbf{b}$, where, for Algorithm 2 at level 7, $\mathbf{x} = (\tilde{\varphi}_m(0),\ldots,\tilde{\varphi}_m(5))$, and $\mathbf{b} = (b_m(0),\ldots,b_m(5))$. The matrix $\mathbf{B}$ is presented in Figure 3.1.

$$\begin{array}{ccccccc}
n & 0 & 1 & 2 & 3 & 4 & 5 \\
0 & 1-\sum_{n=2}^{7}\Gamma_n^2 & -2\sum_{n=2}^{6}\Gamma_n\Gamma_{n+1} & -2\sum_{n=2}^{5}\Gamma_n\Gamma_{n+2} & -2\sum_{n=2}^{4}\Gamma_n\Gamma_{n+3} & -2\sum_{n=2}^{3}\Gamma_n\Gamma_{n+4} & -2\sum_{n=2}^{2}\Gamma_n\Gamma_{n+5} \\
1 & -\Gamma_2 & 1-\Gamma_3 & -\Gamma_4 & -\Gamma_5 & -\Gamma_6 & -\Gamma_7 \\
2 & -\Gamma_3 & -(\Gamma_2+\Gamma_4) & 1-\Gamma_5 & -\Gamma_6 & -\Gamma_7 & 0 \\
3 & -\Gamma_4 & -(\Gamma_3+\Gamma_5) & -(\Gamma_2+\Gamma_6) & 1-\Gamma_7 & 0 & 0 \\
4 & -\Gamma_5 & -(\Gamma_4+\Gamma_6) & -(\Gamma_3+\Gamma_7) & -\Gamma_2 & 1 & 0 \\
5 & -\Gamma_6 & -(\Gamma_5+\Gamma_7) & -\Gamma_4 & -\Gamma_3 & -\Gamma_2 & 1
\end{array}$$

Figure 3.1. The Matrix, B, for the system of equations at level 7

Lemma 3.3. The solution to the system of equations specified by (3.12) through (3.14) will result in a lower bound on the exact value of $\tilde{\varphi}_m(0)$, when M > L.

Proof: The original system of equations (equations 3.9a and 3.9b) is in the form $A\tilde{\varphi}_m = \mathbf{b}_m$, where A is an M–1 x M–1 matrix consists of terms involving Γ_ns. It follows easily, from Lemma 3.1, that the matrix A is an *M*-matrix, i.e., a matrix that is diagonal-dominant, with it's off-diagonal elements non-positive. It is well known (Fiedler and Ptak 1962) that the inverse of such a matrix is positive. Let the inverse be denoted by A^{-1}. We also know that the vector $\mathbf{b}_m$ is positive, and so we know that the true solution $\tilde{\varphi}_m = A^{-1}\mathbf{b}_m > 0$.

Note that the matrix is an (M–1) x (M–1) square matrix and that it's row indices are 0,...,M–2. Let us denote the true solution by the vector $\binom{\mathbf{x}}{\mathbf{y}}$, where **x** denotes the first L–1 elements $\tilde{\varphi}_m(0)$ through $\tilde{\varphi}_m(L-2)$, and **y** denotes the elements $\tilde{\varphi}_m(L-1)$ through $\tilde{\varphi}_m(M-2)$. Analgously, let $\binom{\mathbf{b}^1}{\mathbf{b}^2}$ denote the vector $\mathbf{b}_m$, obtained from equation (3.10), where $\mathbf{b}^1$ denotes elements $b_m(0)$ through $b_m(L-2)$, and $\mathbf{b}^2$ denotes elements $b_m(L-2)$ through $b_m(M-2)$. Denote by **B**, the square submatrix consisting of the first L–1 columns and rows of A, and let **C** denote the matrix consisting of all the remaining elements in the first L–1 rows of **A**. Thus, $\mathbf{Bx} + \mathbf{Cy} = \mathbf{b}^1$. Since matrix **A** is an *M*-matrix, we know that the matrix **C** is nonpositive, and so we must have $\mathbf{Bx} = \mathbf{b}^1 + (-\mathbf{Cy}) > \mathbf{b}^1$. Since we know that $\mathbf{B}^{-1}$ is positive (because $\mathbf{A}^{-1}$ is positive) the solution to $\mathbf{x} = \mathbf{B}^{-1}\mathbf{b}^1$ represents a lower bound on the true solution. Furthermore, equation (3.13) only computes a lower bound (say, **b**,) on the true value of $\mathbf{b}^1$. Thus, the resulting **x** is a lower bound on the exact solution. ■

For L = min(M,7), the expression for $\tilde{\varphi}_m(0)$ can be obtained in a more explicit form as follows. We first exploit the structure of matrix **B**, to express $\tilde{\varphi}_m(i)$, i=2, ... ,5, only in terms $\tilde{\varphi}_m(0)$ and $\tilde{\varphi}_m(1)$ as follows:

$$\tilde{\varphi}_m(i) = u_{m,i} + v_i\,\tilde{\varphi}_m(0) + w_i\,\tilde{\varphi}_m(1), \quad i = 2,\ldots,5, \tag{3.15}$$

where

$$u_{m,2} = \left(b_m(2) + b_m(3)\frac{\Gamma_6+\Gamma_2\Gamma_7}{1-\Gamma_7} + b_m(4)\Gamma_7\right)/\Delta, \qquad u_{m,3} = \frac{b_m(3) + u_{m,2}(\Gamma_2 + \Gamma_6)}{1-\Gamma_7}$$

$$u_{m,4} = b_m(4) + u_{m,2}(\Gamma_3+\Gamma_7) + u_{m,3}\Gamma_2, \qquad u_{m,5} = b_m(5) + u_{m,2}\Gamma_4 + u_{m,3}\Gamma_3 + u_{m,4}\Gamma_2, \tag{3.16a}$$

$$v_2 = \left(\Gamma_3 + \Gamma_4\frac{\Gamma_6+\Gamma_2\Gamma_7}{1-\Gamma_7} + \Gamma_5\Gamma_7\right)/\Delta, \qquad v_3 = \frac{\Gamma_4 + v_2(\Gamma_2 + \Gamma_6)}{1-\Gamma_7}$$

$$v_4 = \Gamma_5 + v_2(\Gamma_3+\Gamma_7) + v_3\Gamma_2, \qquad v_5 = \Gamma_6 + v_2\Gamma_4 + v_3\Gamma_3 + v_4\Gamma_2, \tag{3.16b}$$

$$w_2 = (\Gamma_2+\Gamma_4+(\Gamma_3+\Gamma_5)\frac{\Gamma_6+\Gamma_2\Gamma_7}{1-\Gamma_7} + \Gamma_7(\Gamma_4+\Gamma_6))/\Delta, \quad w_3 = \frac{\Gamma_3+\Gamma_5 + w_2(\Gamma_2+\Gamma_6)}{1-\Gamma_7}$$

$$w_4 = \Gamma_4+\Gamma_6 + w_2(\Gamma_3+\Gamma_7) + w_3\Gamma_2, \qquad w_5 = \Gamma_5+\Gamma_7+w_2\Gamma_4+w_3\Gamma_3+w_4\Gamma_2, \tag{3.16c}$$

and

$$\Delta \;=\; 1-\Gamma_5-(\Gamma_2+\Gamma_6)\frac{\Gamma_6+\Gamma_2\Gamma_7}{1-\Gamma_7} - \Gamma_7(\Gamma_3+\Gamma_7). \tag{3.17}$$

We now express $\tilde{\varphi}_m(1)$ in terms of $\tilde{\varphi}_m(0)$, (refer to the second row of Figure 3.1), and this gives:

$$\tilde{\varphi}_m(1) \;=\; \frac{c_m + a_0}{a_1}, \qquad \text{where} \tag{3.18}$$

$$c_m \;=\; b_m(1) + \Gamma_4 u_{m,2} + \Gamma_5 u_{m,3} + \Gamma_6 u_{m,4} + \Gamma_7 u_{m,5}, \tag{3.19a}$$

$$a_0 \;=\; \Gamma_2 + \Gamma_4 v_2 + \Gamma_5 v_3 + \Gamma_6 v_4 + \Gamma_7 v_5, \qquad \text{and} \tag{3.19b}$$

$$a_1 \;=\; 1-(\Gamma_3 + \Gamma_4 w_2 + \Gamma_5 w_3 + \Gamma_6 w_4 + \Gamma_7 w_5). \tag{3.19c}$$

From equations (3.15) through (3.19c), we obtain the solution for $\tilde{\varphi}_m(0)$ presented below, after some simplification, as Lemma 3.4.

Lemma 3.4. For Algorithm 2 at level 7, $\varphi_m(0)$ is obtained as

$$\varphi_m(0) \;=\; \sum_{j=1}^{M-1} \chi_m^2(j) + \frac{b_m(0) - 2\left(b_m(1)\Gamma_2 + b_m(2)\Gamma_3 - b_m(3)\frac{\Gamma_4\Gamma_7}{1-\Gamma_7} - u_{m,2}\, v_2\,\Delta - c_m\frac{a_0}{a_1}\right)}{1-\sum_{n=2}^{7}\Gamma_n^2 + 2\left(\Gamma_2^2+\Gamma_3^2 - \frac{\Gamma_4^2\Gamma_7}{1-\Gamma_7} - v_2^2\Delta - \frac{a_0^2}{a_1}\right)}. \tag{3.20}$$

We can, of course, simplify the denominator of equation (3.20) somewhat further. We have left it in the current form to indicate how the numerator and denominator are related. (If we replace $b_m(n)$ by Γ_{n+1} for all the terms within the paranthesis in the numerator we obtain the terms within the paranthesis in the denominator of the equation.)

If $M < 7$, then some of these terms are not computed. For example, if $M = 4$, then $\Delta = 1$, and we only compute $u_{m,i}$, v_i and w_i for i=2, and set the terms involving Γ_5, Γ_6 and $\Gamma_7 = 0$, in equation (3.20). Similarly, for M=3, $u_{m,i}$, v_i, and $w_i = 0$ for all i, $c_m = b_m(1)$, $a_0 = \Gamma_2$, $a_1 = 1-\Gamma_3$, and $b_m(2) = 0$ (note that k ranges only from 1 through M−2 in equation 3.10). Thus, for M=3:

$$\tilde{\varphi}_m(0) \;=\; \frac{b_m(0) + 2b_m(1)\Gamma_2\Gamma_3/(1-\Gamma_3)}{1-\Gamma_2^2-\Gamma_3^2-2\Gamma_2^2\Gamma_3/(1-\Gamma_3)}. \tag{3.21}$$

From equations (3.10) and (3.11), $b_m(0) = g_m^2(1) + g_m^2(2) = (\chi_m(2)\Gamma_2 + \chi_m(1)\Gamma_3)^2 + (\chi_m(2)\Gamma_3)^2$, and $b_m(1) = \chi_m(2)\,(\chi_m(2)\Gamma_2 + \chi_m(1)\Gamma_3)$. If we substitute these values in equation (3.21), we obtain the same expression for $\tilde{\varphi}_m(0)$ as in equation (3.8a).

Algorithm 2:

0. Initialization: Compute a) δ_m, $m=1,\ldots,M$ (equation 2.10); b) Γ_n, $n = 2,\ldots,L$, (equation (3.3); c) Δ (equation 3.17); d) v_i and w_i, $i = 2, \ldots, L-2$, (equations 3.16b and 3.16c); and e) a_0 and a_1 (equations 3.19b and 3.19c). Evaluate the denominator of equation (3.20).
1. For every node, k, whose mean waiting time is desired, renumber the nodes so that node k is node "1", node k+1 is node "2", ..., and execute steps 2 and 3.
2. For each $m=1,\ldots,M$, do steps a) through d).
   a) Compute $\chi_m(j)$, $j = 1,\ldots,M-1$, (equation 3.1).
   b) Compute $b_m(j)$, $j = 1,\ldots,L-2$, using (equation 3.13).
   c) Compute $u_{m,i}$, $i=2,\ldots,L-2$, (equation 3.16a), and c_m (equation 3.19a).
   d) Compute $\varphi_m(0)$ (equation 3.20).
3. Obtain the mean waiting time using equation (2.25).

The computational complexity of Algorithm 2 is obtained as follows. Evaluating Γ_n requires $O(M^2)$ operations, regardless of the level L. The rest of the computations in Step 0 requires O(1) time. These computations are incurred regardless of the number of nodes being evaluated. Step 2 requires $O(M^2)$ operations each time it is carried out. The major computational effort in this step arises from steps 2a) and 2d). Hence, the overall computational effort is $O(M^2)$ for each node that is evaluated for its mean waiting time.

In terms of storage requirements, the algorithm only needs one 2-dimensional array of size $M-1 \times M-1$ to store the $\chi_m(j)$ terms, and several one-dimensional arrays of size M or less.

Computational Remark: If M is large, then there can be significant savings in computation if we make use of equations (3.15) and (3.18) as follows. From equation (3.1), we obtain

$$\begin{aligned}
\tilde{\varphi}_{m+1}(0) &= \sum_{j=M}^{\infty} \chi_{m+1}^2(j) = \rho_{m+1}^2 \sum_{j=M}^{\infty} \left(\frac{\chi_m(j)}{\delta_m} + \chi_m(j-1)\right)^2 \\
&= \rho_{m+1}^2 \left(\frac{\tilde{\varphi}_m(0)}{\delta_m^2} + \tilde{\varphi}_m(0) + \chi_m^2(M-1) + 2\sum_{j=M}^{\infty} \frac{\chi_m(j)\chi_m(j-1)}{\delta_m} \right) \\
&= \rho_{m+1}^2 \left(\tilde{\varphi}_m(0)\left(1 + \frac{1}{\delta_m^2}\right) + 2\frac{\tilde{\varphi}_m(1)}{\delta_m} + \chi_m^2(M-1) \right). \qquad (3.22)
\end{aligned}$$

The term $\tilde{\varphi}_m(1)$ is obtained, from equation (3.18), as $\tilde{\varphi}_m(1) = (c_m + a_0\tilde{\varphi}_m(0))/a_1$. Thus, once we compute $\tilde{\varphi}_m(0)$, we directly obtain $\tilde{\varphi}_{m+1}(0)$ with just a few arithmetic operations, from equations (3.18) and (3.22). We can carry this approach further as follows. The term $\tilde{\varphi}_{m+1}(1)$ is obtained, similar to equation (3.22), as

$$\tilde{\varphi}_{m+1}(1) = \rho_{m+1}^2\left(\tilde{\varphi}_m(1)(1+\frac{1}{\delta_m^2})+\frac{\tilde{\varphi}_m(0)+\tilde{\varphi}_m(2)}{\delta_m}+\chi_m(M-1)[\chi_m(M-1)/\delta_m+\chi_m(M-2)]\right). \quad (3.23)$$

We compute $\tilde{\varphi}_m(2)$ from equation (3.15), and use it in equation (3.23) to evaluate $\tilde{\varphi}_{m+1}(1)$. Having computed $\tilde{\varphi}_{m+1}(0)$ and $\tilde{\varphi}_{m+1}(1)$, we obtain $\tilde{\varphi}_{m+2}(0)$ using equation (3.22) once again. It is, of course, possible to carry this idea further. Thus, we can compute $\tilde{\varphi}_{m+3}(0)$ if we know $\tilde{\varphi}_{m+2}(0)$ and $\tilde{\varphi}_{m+2}(1)$. Note that, in order to compute $\tilde{\varphi}_{m+2}(1)$, we must compute $\tilde{\varphi}_{m+1}(2)$, and so on. The savings in computation become insignificant after some time. In our implementation of Algorithm 2, we compute $\tilde{\varphi}_m(0)$, for m = 1, 4, 7, ..., from equation (3.20), and then compute $\tilde{\varphi}_m(0)$ for m = 2, 3, 5, 6, 8, 9, ... using the above approach. We can show that this approach still results in a lower bound on the exact mean waiting times. ■

The next section on computational experience demonstrates the accuracy of Algorithm 2 at level 7. We remind the reader that Algorithm 2 at level 7 obtains the exact mean waiting times if $M \le 7$.

4. Computational Experience

We conducted a number of experiments with randomly generated networks, to test the performance of the two algorithms. We were interested in determining a) the execution times for Algorithm 1, and b) the execution times and accuracy of Algorithm 2. We varied the number of nodes from 3 to 50, and the server utilization, ρ, from 0.30 to 0.996. We report on some experimental results in this section, which appear to best represent the performance of these algorithms. The code was implemented in Pascal and the experiments were conducted on an IBM 9021-720 running the MTS operating system.

The iterative algorithm was terminated as soon as the largest $\chi_m(j)$ value obtained during an iteration dropped below 0.0001. Based on extensive comparisons with exact results, we found that this termination criterion gave accurate results. (The results were accurate upto 4 decimal places, and resulted in a percentage error of less than 0.0001%.) We observed that Algorithm 1 typically ran very quickly when the value of ρ was less than 0.90. We also observed that Algorithm 2 was uniformly extremely accurate and obtained mean waiting times very quickly. We ran Algorithm 2 at level 6 for $\rho \le 0.90$, and this always gave the exact results (accurate upto 6 decimal places). For $\rho = 0.99$, we ran Algorithm 2 at level 7, and the mean waiting times obtained from Algorithm 2, when compared to the corresponding exact mean waiting times, had a maximum percentage deviation of *less than 0.0025%* for all the experiments we conducted. At this point, we must emphasize that if even greater accuracy is required, then we only have to set $L = \min(M,8)$, and solve the "level 8" algorithm (this will involve the following additional computations: a one-time computation of the terms v_6 and w_6, and subsequent computations of $b_m(6)$ and $u_{m,6}$).

In Table 1, we report on the maximum number of iterations and the running time of Algorithm 1, and the accuracy and running times for Algorithm 2. The results are reported for the following values of ρ: 0.30, 0.50, 0.75, 0.90, and 0.99, for $M = 12, 24, 48$ and 96 nodes. For each example considered, the algorithms were executed to obtain the mean waiting time for: a) 25% of the nodes, b) 50% of the nodes, c) 75% of the nodes, and d) 100% of the nodes. As expected, the execution time for Algorithm 1 increased in a linear manner with the number of nodes that were evaluated for mean waiting times. The execution time for Algorithm 2 was measured as the sum of two components: the time taken to complete the initialization phase, and the time taken for the second phase, namely to compute the mean waiting times. As expected, the initialization time, for a given ρ and M, was the same regardless of the number of nodes evaluated while the execution time for the second phase grew in a linear manner with the number of nodes evaluated (subject to insignificant variations in CPU times). Table 1 reports the execution times for these two phases.

Table 1 indicates the growth of the execution times for the two algorithms grow with increasing values for ρ and M. Interestingly enough, Algorithm 1 uniformly appears to take roughly the same number of iterations for a given ρ, regardless of the number of nodes. In general, Algorithm 1 performs very well, so long as ρ is about 0.75 or less. For higher values of ρ, the performance of Algorithm 1 steadily degrades. Since it consistently takes about 28 to 35 iterations for $\rho = 0.90$, and since each iteration requires $O(M)$ operations, this suggests that if $\rho < 0.90$ and M is larger than around 40, Algorithm 1 would require less than $O(M^2)$ operations to compute $\varphi_m(0)$, $m=1,\ldots,M$, and, therefore, require less than $O(M^2)$ operations to find the mean waiting time at a node. This also suggests that if Algorithm 1 is used to compute all the mean waiting times, it would probably outperform an $O(M^3)$ algorithm that computes all the mean waiting times, for large values of M (say, for $M > 40$), and $\rho < 0.90$.

Algorithm 2, of course, does not depend on the value of ρ, but on the other hand, it requires more time for increasing values of M. As far as the accuracy of Algorithm 2 is concerned, it never gave an error greater than 0.0022%. We emphasize that this is a *percentage difference*, namely that the relative error is multiplied by 100. We ran Algorithm at level 6 for $\rho < 0.99$, and it gave the exact values for the mean waiting times (correct to 4 decimal places) every time. (This explains the marginal increase in the execution time observed for $\rho = 0.99$.) As expected, from Table 1, the observed growth in CPU time required by Algorithm 2 is of $O(M^3)$to compute the mean waiting times at all nodes , and of $O(M^2)$ to compute the mean waiting time at a single node.

Table 1. Performance of Algorithms 1 and 2.

M	ρ	Algorithm 1			Algorithm 2			
		Max. # iterations	Tot. cpu time (ms)	cpu time/ node (ms)	Max. error (%)	Tot. cpu time(ms)[1]	Initializn. time(ms)[2]	cpu time/ node(ms)[3]
	0.30	5	10.04	0.84	0.0000	15.62	2.55	1.09
	0.50	7	11.68	0.97	0.0000	15.67	2.61	1.09
12	0.75	14	17.60	1.47	0.0000	15.64	2.54	1.09
	0.90	35	35.71	2.98	0.0000	15.54	2.64	1.08
	0.99	368	316.55	41.66	0.0005	16.43	2.60	1.15
	0.30	5	48.51	2.02	0.0000	95.53	10.17	3.56
	0.50	7	55.54	2.31	0.0000	95.20	10.33	3.54
24	0.75	13	76.00	3.17	0.0000	94.79	10.14	3.53
	0.90	34	146.87	6.12	0.0000	95.48	10.28	3.55
	0.99	356	1,242.42	51.77	0.0022	99.76	10.37	3.72
	0.30	4	275.48	5.74	0.0000	675.94	60.71	12.82
	0.50	6	305.16	6.36	0.0000	675.41	60.74	12.81
48	0.75	13	392.82	8.18	0.0000	673.31	60.62	12.76
	0.90	31	649.13	13.52	0.0000	676.33	60.74	12.82
	0.99	331	4,738.71	98.72	0.0018	691.15	61.28	13.12
	0.30	4	1,894.48	19.73	0.0000	5,081.83	437.11	48.38
	0.50	6	2,002.55	20.86	0.0000	5,106.79	438.07	48.63
96	0.75	11	2,293.96	23.90	0.0000	5,080.99	440.98	48.33
	0.90	28	3,235.45	33.70	0.0000	5,127.91	439.43	48.84
	0.99	302	18,212.95	189.72	0.0000	5,158.67	440.15	49.15

We present some examples to give the reader some feel for actual numerical values. These examples also serve to illustrate the accuracy of Algorithm 2 at level 7, since all of them have very large ρ values. In one example with 12 nodes, we set $\rho = 0.996$. The percentage error for the largest deviation found in this example, was equal to 0.0016%.

[1] Algorithm 2 was always run at level 6 for $\rho < 0.99$.
[2] Initialization Time is incurred regardless of the number of nodes evaluated for their mean waiting times.
[3] Excluding the initialization time

Example 1. The number of nodes = 5, with arrival rates 0.2, 0.8, 0.4, 0.2, and 0.1. The mean service times are 0.5, 0.5, 0.4, 1.0 and 1.2, and the second moments of the service times are 0.5, 0.5, 0.3, 2.0, and 2.9. The mean switchover times are 0.5, 0.5, 0.4, 1.0, and 1.2. The variances of the switchover times are equal to their means. For this problem, $\rho = 0.98$.

The mean waiting times are $W_1 = 121.0880$, $W_2 = 80.7446$, $W_3 = 113.3191$, $W_4 = 107.7545$, $W_5 = 118.3033$.

Example 2. The number of nodes = 7, with arrival rates 0.2, 0.6, 0.4, 0.2, 0.1, 0.1, and 0.1. The service times have means 0.5, 0.5, 0.4, 1.0, 0.6, 0.6, and 1.1, and second moments 0.3, 0.3, 0.2, 1.5, 0.8, 0.8, and 1.8. The mean switchover times are 0.5, 0.5, 0.8, 1.0, 0.6, 0.6, and 1.1. The variances of the switchover times are equal to their means. For this problem, $\rho = 0.99$.

The mean waiting times are $W_1 = 283.0562$, $W_2 = 220.2503$, $W_3 = 264.4854$, $W_4 = 251.7250$, $W_5 = 295.7516$, $W_6 = 295.7251$, $W_7 = 279.8502$.

Example 3. The number of nodes = 10, with arrival rate equal to 0.75 at node 1, and 0.2 at all the other nodes. The mean service times are $b_1 = 0.50$, $b_2 = \ldots = b_5 = 0.40$, $b_6 = b_7 = b_8 = 0.3$, and $b_9 = b_{10} = 0.2$. The second moments of the service times are as follows: 0.5 at node 1, 0.3 at nodes 2 through 5, 0.2 at nodes 6 through 8, and 0.05 at nodes 9 and 10. The mean switchover times are 0.5 at node 1, 0.2 at nodes 2 through 5, 0.3 at nodes 6 through 8, and 0.5 at nodes 9 and 10. The variances of the switchover times are 0.25 at node 1, 0.15 at nodes 2 through 5, 0.20 at nodes 6 through 8, and 0.25 at nodes 9 and 10. The value of $\rho = 0.955$.

The mean waiting times obtained by Algorithm 2 matched the exact mean waiting times (correct to 4 decimal places). The mean waiting times are $W_1 = 28.8749$, $W_2 = 42.5150$, $W_3 = 42.5305$, $W_4 = 42.5481$, $W_5 = 42.5684$, $W_6 = 43.5228$, $W_7 = 43.5544$, $W_8 = 43.5905$, $W_9 = 44.5781$, and $W_{10} = 44.6425$.

Example 4. The number of nodes = 12, with $\lambda_i = 0.01\, i$, $i = 1,\ldots,12$. The mean (second moment) of the service time is equal to 1.60 (2.00) at odd-numbered nodes. The mean (second moment) of the service time is equal to 1.00 (1.50) at even-numbered nodes. The switchover times have identical means and identical variances at all nodes, and they are, respectively, 0.50 and 0.25. The ρ value for this example is 0.996.

(Algorithm 1 required over 1200 iterations to converge to the exact values.) Comparing the exact mean waiting times with the mean waiting times obtained by Algorithm 2, the largest deviation from the exact mean waiting time occurred at node 1. At node 1, the exact mean waiting time is 924.8319. The corresponding mean waiting time obtained by Algorithm 2 was 924.8174. This resulted in an absolute deviation of 0.0145, and a percentage deviation of 0.0016%.

Example 5. The number of nodes = 24. $\lambda_i = 0.4$ for nodes 1 and 13, $\lambda_i = 0.20$ for nodes 2 and 14. For all other nodes, $\lambda_i = 0.01$. The mean (second moment) of the service time is equal to 0.80 (2.00) at odd-numbered nodes, and equals 0.50 (1.50) at even-numbered nodes. The mean switchover times are all equal to 0.50, and the variances of the switchover times are all equal to 0.25. The value of $\rho = 0.97$.

The mean waiting times are 173.8729 at nodes 1 and 13, and 230.1439 at nodes 2 and 14. The mean waiting times at nodes 3 through 12 range from 253.6835 to 254.5273. The mean waiting times at nodes i and i+14 are identical (upto 4 decimal places) for i=1,...,10. The mean waiting times obtained by Algorithm 2 matched the exact mean waiting times upto 4 decimal places.

Example 6. The number of nodes = 48. $\lambda_i = 1.50$ for nodes 1 and 2, 1.00 for nodes 3 and 4, 0.50 for nodes 5 through 10, 0.45 for nodes 11 through 22, 0.30 for nodes 23 through 34, and 0.20 for all other nodes. The mean (second moment) of the service time is 0.05 (0.05) at all nodes. The mean and variance of the switchover times are equal to 0.10 and 0.08, respectively, at all nodes. The value of $\rho = 0.99$.

The exact mean waiting times at nodes 1 through 4 were $W_1 = 269.6124$, $W_2 = 269.6075$, $W_3 = 276.8928$, and $W_4 = 276.8912$. At nodes 5 through 10, W_i ranged from 284.1779 to 284.1786. At nodes 11 through 22, W_i ranged from 284.9075 to 284.9103. At nodes 23 through 34, W_i ranged from 287.0966 to 287.1007. At nodes 35 through 48, the mean waiting times ranged from 288.5583 to 288.5627. The mean waiting times obtained by Algorithm 2 were compared with the exact mean waiting times. The largest absolute deviation from the exact mean waiting time occurred at node 34. The exact mean waiting time at node 34 was 287.1007, while the mean waiting time obtained by Algorithm 2 was 287.0962, resulting in an absolute error of 0.0045, and a percentage deviation of about 0.0016%.

In all these examples, the mean waiting times for nodes with identical characteristics are very nearly equal. This indicates that for systems where many nodes have similar characteristics (as is usually the case for large systems), it may be more appropriate, at least during the design stage of the network when many alternate configurations may need to be considered, to evaluate mean waiting times only for some select few nodes. For instance, in example 7, it may be adequate to evaluate the mean waiting times at nodes 1, 3, 5, 23, and 35. It is especially in such situations that the IS technique has a distinct advantage over other techniques where all nodes have to be evaluated simultaneously.

4.1. Comparison with the algorithm of Sarkar and Zangwill

We compared the execution times for Algorithms 1 and 2 with the execution time for Sarkar and Zangwill's algorithm (the S&Z algorithm). Before we present the results, we make the disclaimer that while we tried to implement the S&Z algorithm as efficiently as possible, there is always a possibility that one could improve its implementation even further. The same disclaimer holds for the implementations of Algorithms 1 and 2. (Note that if Algorithm 2 is used to compute the mean waiting times at all nodes, it is an $O(M^3)$ algorithm, as is the S&Z algorithm.) We ran the S&Z algorithm using the same data that was used to generate Table 1. The algorithm was run on the same IBM 9021-720. The S&Z algorithm was coded in Fortran, which made it easier to call LINPACK (a library of subroutines for solving linear systems of equations, written in Fortran).

The execution times for the three algorithms are presented graphically in Figures 4.1 through 4.4, for $M = 12, 24, 48$, and 96, respectively. These figures show the increase in execution times of the Algorithms 1 and 2, with the number of nodes that are evaluated for their mean waiting times. Since the execution time of Algorithm 1 increases with ρ, for each M, we present the execution times of Algorithm 1 for $\rho = 0.50$, 0.75 and 0.90. We do not present the execution time of Algorithm 1 at $\rho = 0.99$, since it is relatively very large. Unlike Algorithm 1, Algorithm 2 and the S&Z algorithm do not depend on ρ. Therefore we do not present the execution times of Algorithm 2 and the S&Z algorithm for different ρ values; the execution times presented are the averages of the execution times for the different ρ values.

Based on these figures, Algorithm 2 would be preferred over the S&Z algorithm if mean waiting times are desired at less than about 60% of the nodes. Algorithm 2 takes about 60% more time than the S&Z algorithm, to evaluate all the mean waiting times, using a single computer. However, it is noted that Algorithm 2, following the initialization phase, computes the mean waiting times for individual nodes independently. Therefore, using Algorithm 2, one could obtain mean waiting times at all nodes faster than the S&Z algorithm if another process was spawned on a similar computer, once the initialization phase ends. The two processes could work in parallel, each computing mean waiting times at, say, half the nodes, reducing the execution time for the second phase of Algorithm 2 by half. The developments in distributed computing makes such an approach very feasible.

The performance of Algorithm 1 steadily improves (for $\rho \leq 0.90$) as the number of nodes increase, relative to the performance of the other two algorithms. For $M = 96$ nodes, Algorithm 1 computes all the mean waiting times faster than the S&Z algorithm even at $\rho = 0.90$. This agrees with an earlier observation we made about the relative efficiency of Algorithm 1 as compared to an $O(M^3)$ algorithm for large M and ρ less than 0.90.

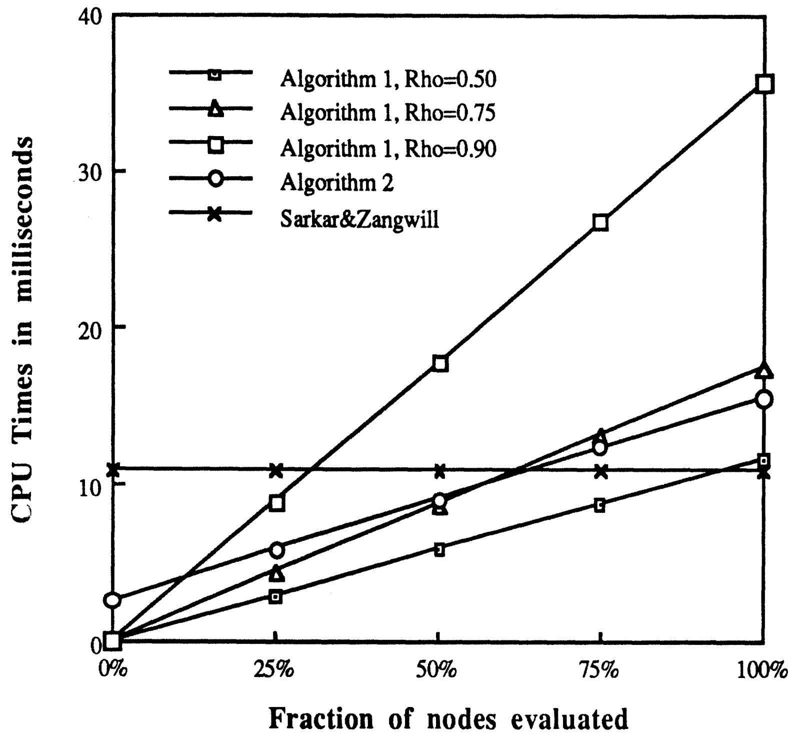

Figure 4.1: Execution times for M = 12 nodes

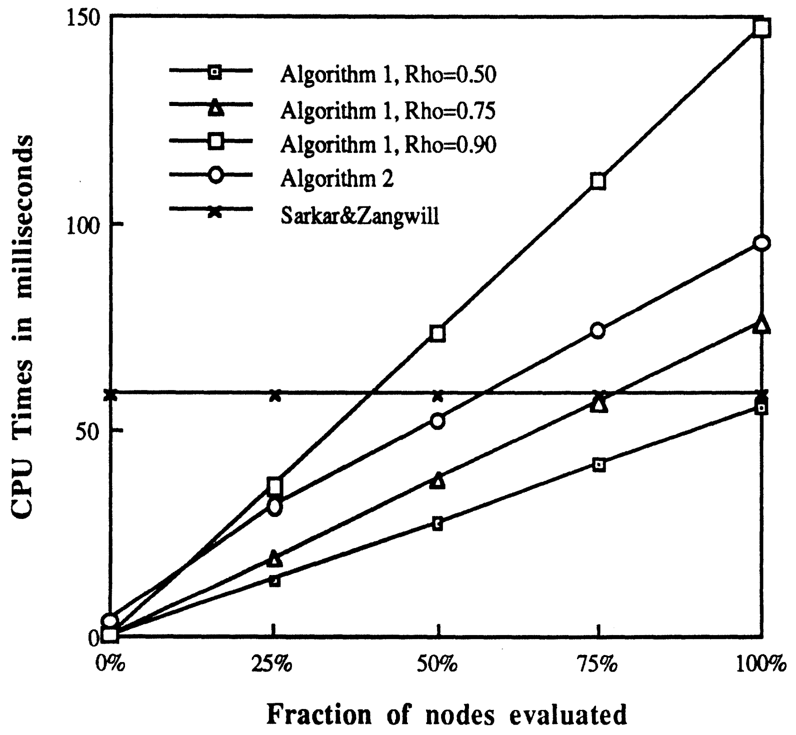

Figure 4.2: Execution times for M = 24 nodes

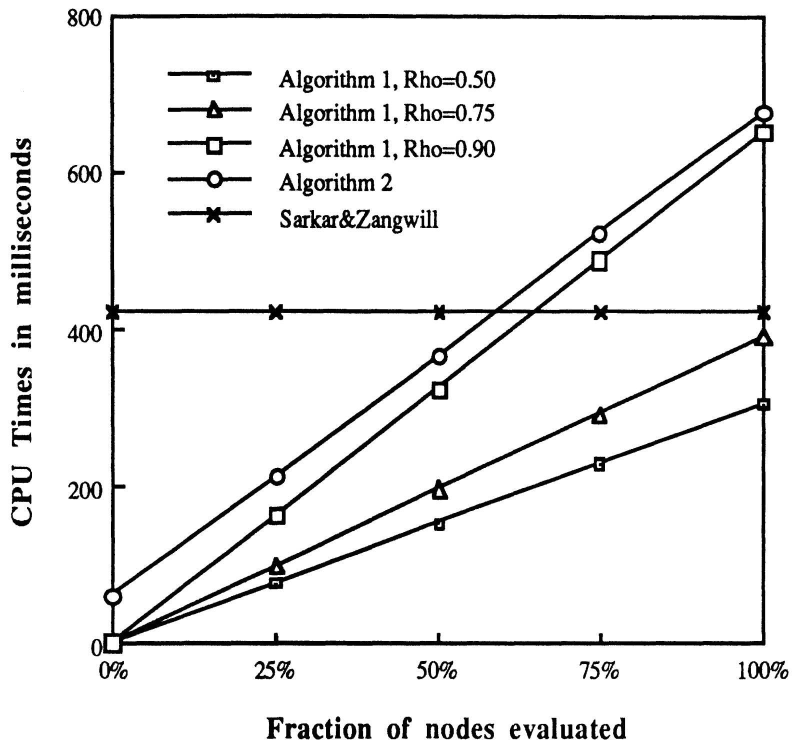

Figure 4.3: Execution times for M = 48 nodes

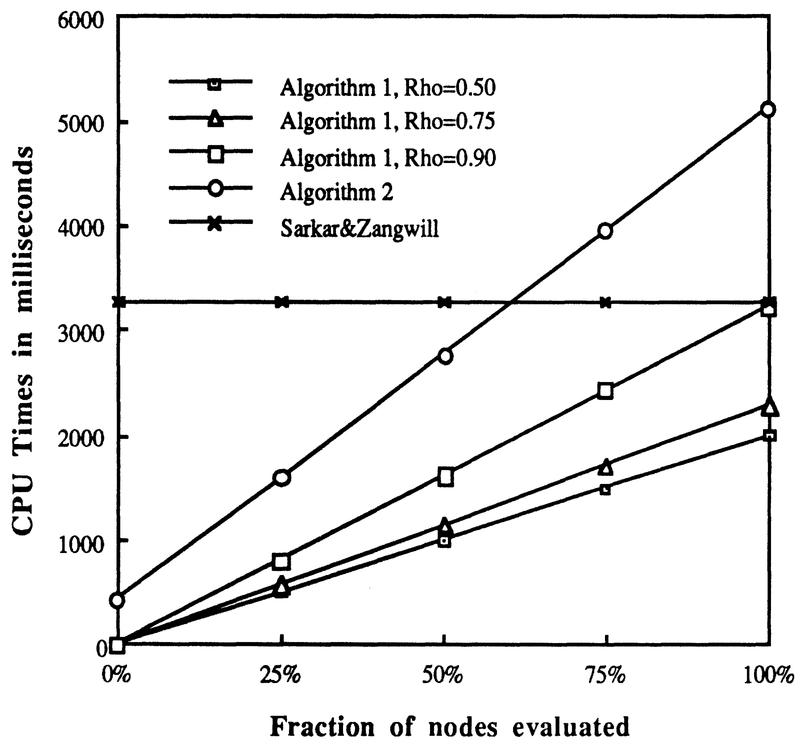

Figure 4.4: Execution times for M = 96 nodes

5. The Gated Service Polling System

The analysis for the gated service polling system is very similar to the analysis for the exhaustive service polling system. As before, we obtain the mean waiting times at the nodes, after computing the first and second factorial moments of the number of customers present at the node when the server polls it. The mean waiting time is given in terms of f_m and $f_m^{(2)}$ as (Takagi 1986):

$$W_m = \frac{f_m^{(2)}}{2f_m}\,\frac{1+\rho_m}{\lambda_m}. \tag{5.1}$$

As before, we first obtain f_1 and $f_1^{(2)}$. To compute these terms, we require a redefinition of the recursive funtion γ as follows:

$$\gamma_m(0) = z_m, \qquad m = 1,\ldots,M, \tag{5.2a}$$

$$\gamma_m(j) = \beta_m\Big(\sum_{k\le m}[\lambda_k - \lambda_k\gamma_k(j-1)] + \sum_{k>m}[\lambda_k - \lambda_k\gamma_k(j)]\Big), \quad j>0, \quad m=1,\ldots,M. \tag{5.2b}$$

The term $\sigma_m(j)$ is still defined by equation (2.5), with the redefined γ_m terms. Proceeding exactly as before, we obtain:

$$F_1(\gamma_1(0),\ldots,\gamma_M(0)) = \prod_{j=0}^{\infty}\Big(\prod_m \sigma_m(j)\Big) \tag{5.3}$$

Equations (2.8), (2.9), and (2.12) still hold. Equation (2.11), however, is modified as follows:

$$\psi_m(j) = \rho_m\Big[\sum_{k\le m}\psi_k(j-1) + \sum_{k>m}\psi_k(j)\Big], \quad j>0, \text{ and} \tag{5.4a}$$

$$\psi_m^{(2)}(j) = \rho_m\Big[\sum_{k\le m}\psi_k^{(2)}(j-1) + \sum_{k>m}\psi_k^{(2)}(j)\Big] + \lambda_m b_m^{(2)}\Big[\sum_{k\le m}\psi_k(j-1) + \sum_{k>m}\psi_k(j)\Big]^2, \quad j>0. \tag{5.4b}$$

Following a similar analysis as used to obtain f_1 for the exhaustive service case, we get

$$f_1 = \lambda_1\frac{s}{1-\rho}. \tag{5.5}$$

The term $f_1^{(2)}$ is given by the same expression as for the exhaustive service case:

$$f_1^{(2)} = f_1^2 + \sum_m s_m \sum_k y_k + \sum_m \mathrm{Var}(S_m)\sum_{j=0}^{\infty}\Big[\sum_{k\le m}\psi_k(j) + \sum_{k>m}\psi_k(j+1)\Big]^2, \tag{5.6}$$

where

$$\sum_m y_m = \sum_m \frac{\lambda_m b_m^{(2)}}{(1-\rho)\rho_m^2}\sum_{j=1}^{\infty}\psi_m^2(j). \tag{5.7}$$

However, unlike the exhaustive service case, where $[\sum_{k\leq m}\psi_k(j)+\sum_{k>m}\psi_k(j+1)]^2$ depended on node m+1 (refer to equation 2.20), for the gated service case we observe, from equation (5.4a), that this term simply evaluates to $\psi_m(j+1)$. Hence, equation (5.6) is rewritten as:

$$f_1^{(2)} = f_1^2 + s\sum_m \frac{\lambda_m b_m^{(2)}}{(1-\rho)\rho_m^2}\sum_{j=1}^{\infty}\psi_m^2(j) + \sum_m \frac{\mathrm{Var}(S_m)}{\rho_m^2}\sum_{j=1}^{\infty}\psi_m^2(j). \tag{5.8}$$

Define, as before, the terms $\chi_m(j)$, $\varphi_m(n)$, and $\tilde{\varphi}_m(n)$ as

$$\chi_m(j) = \frac{\psi_m(j)}{\lambda_1} = \frac{\lambda_m}{\lambda_1}\frac{\partial\gamma_m(j)}{\partial z_1}, \tag{5.9}$$

$$\varphi_m(n) = \sum_{j=n+1}^{\infty}\chi_m(j)\chi_m(j-n), \quad\text{and}\quad \tilde{\varphi}_m(n) = \sum_{j=M}^{\infty}\chi_m(j)\chi_m(j-n). \tag{5.10}$$

The expression for $f_1^{(2)}$ for the gated service polling system is given below as Lemma 5.1.

Lemma 5.1:

$$f_1^{(2)} = f_1^2 + \lambda_1^2\sum_m \varphi_m(0)\left(\frac{s}{1-\rho}\frac{\lambda_m b_m^{(2)}}{\rho_m^2} + \frac{\mathrm{Var}(S_m)}{\rho_m^2}\right). \quad \blacksquare$$

From Lemma 5.1, and equations (5.1) and (5.5), we obtain:

Theorem 5.2: The mean waiting time at node 1 for the gated service polling system is:

$$W_1 = \frac{s}{2}\frac{1+\rho_1}{1-\rho} + (1+\rho_1)\sum_m \frac{\varphi_m(0)}{\rho_m^2}\left(\frac{\lambda_m b_m^{(2)}}{2} + \frac{1-\rho}{2s}\mathrm{Var}(S_m)\right). \tag{5.11}$$

$\blacksquare$

5.1. Computing $\varphi_m(0)$ for the gated service system

As before, we present two algorithms for computing $\varphi_m(0)$. Let

$$\delta_m = \rho_m/(1+\rho_m). \tag{5.12}$$

Similar to the exhaustive service case, the $\chi_m(j)$ terms are computed using the following expression, which is obtained from equations (5.4a) and (5.9):

$$\chi_M(1) = \rho_M; \qquad \chi_m(1) = (\chi_{m+1}(1)/\delta_{m+1})\rho_m,\ \ m=1,\ldots,M-1, \tag{5.13a}$$

and for j > 1,

$$\chi_M(j) = (\chi_1(j-1)/\delta_1 - \chi_1(j-2))\rho_M; \quad \chi_m(j) = (\chi_{m+1}(j)/\delta_{m+1} - \chi_{m+1}(j-1))\rho_m,\ \ m=1,\ldots,M-1. \tag{5.13b}$$

The iterative algorithm to compute $\varphi_m(0)$ is identical to the one we developed for the exhaustive service case, and so we do not repeat it here. The algorithm which computes $\tilde{\varphi}_m(0)$ by solving a system of equations is also very similar to the corresponding one for the exhaustive service case. Some differences, however, exist, which we clarify here. Define the terms $\kappa^{(i)}_m$ and Ω_n as:

$$\kappa^{(1)}_m = \rho_m, \quad m = 1,\ldots,M; \qquad \kappa^{(i)}_m = \rho_m \sum_{j=i-1}^{m-1} \kappa^{(i-1)}_j; \tag{5.14}$$

$$\Omega_M = \prod_m \rho_m, \quad \text{and} \quad \Omega_m = \Omega_{m+1} + \sum_{k=m}^{M} \kappa^{(i)}_m, \quad 1 \le m < M. \tag{5.15}$$

Intuitively, Ω_1, for example, denotes the sum of all possible combinations of unique ρ_m terms taken singly (i.e., $\rho_1+\ldots+\rho_M$) + the sum of all possible combinations of 2 unique ρ_m terms + ... + the sum of all possible combinations of M unique ρ_m terms. The terms, Γ_n are defined as:

$$\Gamma_1 = \Omega_1, \quad \text{and} \quad \Gamma_n = (-1)^{n+1} \sum_{m=n}^{M} \binom{m-2}{n-2} \Omega_m, \quad n = 2,\ldots,M. \tag{5.16}$$

Defining the terms $\theta_m(j)$ as follows:

$$\theta_M(1) = \rho_M, \quad \theta_M(j) = 0, \; j > 1, \tag{5.17a}$$

$$\theta_m(j) = \rho_m(\theta_{m+1}(j)/\delta_{m+1} - \theta_{m+1}(j-1)), \quad m > 1, \; j = 1,\ldots,M-1, \tag{5.17b}$$

$$\theta_1(j) = -\textstyle\sum_{m=2}^{M} \theta_m, \quad j = 1,\ldots,M-1, \tag{5.17c}$$

we can obtain an expression for $\chi_m(j)$ in terms of $\chi_m(j-n)$ stated below as Lemma 5.3. The proof of Lemma 5.3 is by induction on M, and is omitted.

Lemma 5.3

$$\chi_m(j) = \sum_{n=1}^{M} \chi_m(j-n)\, \Gamma_n, \qquad j \ge M, \tag{5.18}$$

with

$$\chi_m(j) = \sum_{n=1}^{j} \chi_m(j-n)\, \Gamma_n + \theta_m(j), \qquad j \le M-1 \tag{5.19}$$

Note that the Γ_n terms in equation (5.18) for the gated service polling system range from n=1, through M, as opposed to the corresponding equation (3.4) for the exhaustive service polling system, where the Γ_n terms range from 2 through M. Based on equation (5.18), we can derive the following system of equations for obtaining $\tilde{\varphi}_m(0)$:

$$\tilde{\varphi}_m(0) = b_m(0) + \tilde{\varphi}_m(0) \sum_{n=1}^{M} \Gamma_n^2 + 2 \sum_{i=1}^{M-1} \tilde{\varphi}_m(i) \sum_{n=1}^{M-i} \Gamma_n \Gamma_{n+i}, \tag{5.20a}$$

$$\tilde{\varphi}_m(k) = b_m(k) + \sum_{n=1-k}^{M-k} \tilde{\varphi}_m(n)\Gamma_{n+k}, \qquad M-1 \geq k > 0, \tag{5.20b}$$

$$b_m(0) = \sum_{n=1}^{M} g_m^2(n), \qquad b_m(k) = \sum_{n=1}^{k} \chi_m(M-1-k+n)\, g_m(n), \quad k = 1,\ldots,M-1, \tag{5.21}$$

where $g_m(n)$ is computed, recursively, from

$$g_m(n) = \sum_{k=n}^{M} \chi_m(M-1-k+n)\, \Gamma_k. \tag{5.22}$$

As before, in equation (5.20b) we use the convention that $\tilde{\varphi}_m(-k) = \tilde{\varphi}_m(k)$.

This system of equations is, once again, of the form $A\varphi_m = \mathbf{b}_m$, and can be solved for the mean waiting times. For the gated service polling system with M nodes, we have M equations in M unknowns, namely, the $\tilde{\varphi}_m(k)$s, $k=0,\ldots,M-1$. As before, the terms Γ_n decrease at an exponential rate with n. However, we were unable to prove that by ignoring terms involving Γ_n, say, for $n \geq 7$, we would obtain a lower bound on the mean waiting time.

For the gated service polling system, we observe that if we only consider terms upto Γ_6, ignoring terms involving Γ_n, for $n \geq 7$ (the resulting solution is termed the solution to Algorithm 2 at "level 6"), excellent results are obtained if we also disregard all Ω_n terms for $n \geq 7$, in computing the Γ's. Lemma 5.4 presents the level 6 system of equations. It may be observed that they are strikingly similar to the level 7 equations for the exhaustive service polling system. Also, note that Algorithm 2 executed at level 6 returns the exact numerical results for a system with upto 6 nodes.

Lemma 5.4. For Algorithm 2 at level 6, $\varphi_m(0)$ is obtained as

$$\varphi_m(0) = \sum_{j=1}^{M-1} \chi_m^2(j) + \frac{b_m(0) - 2\left(b_m(1)\Gamma_1 + b_m(2)\Gamma_2 - b_m(3)\dfrac{\Gamma_3\Gamma_6}{1-\Gamma_6} - u_{m,2}v_2\Delta - c_m\dfrac{a_0}{a_1}\right)}{1 - \sum\limits_{n=1}^{6}\Gamma_n^2 + 2\left(\Gamma_1^2 + \Gamma_2^2 - \dfrac{\Gamma_3^2\Gamma_6}{1-\Gamma_6} - v_2^2\Delta - \dfrac{a_0^2}{a_1}\right)}, \tag{5.23}$$

where

$$c_m = b_m(1) + \Gamma_3 u_{m,2} + \Gamma_4 u_{m,3} + \Gamma_5 u_{m,4} + \Gamma_6 u_{m,5}, \tag{5.24a}$$

$$a_0 = \Gamma_1 + \Gamma_3 v_2 + \Gamma_4 v_3 + \Gamma_5 v_4 + \Gamma_6 v_5, \tag{5.24b}$$

$$a_1 = 1 - (\Gamma_2 + \Gamma_3 w_2 + \Gamma_4 w_3 + \Gamma_5 w_4 + \Gamma_6 w_5), \text{ and} \tag{5.24c}$$

$$\Delta = 1 - \Gamma_4 - (\Gamma_1 + \Gamma_5)\frac{\Gamma_5 + \Gamma_1\Gamma_6}{1-\Gamma_6} - \Gamma_6(\Gamma_2 + \Gamma_6). \tag{5.24d}$$

The terms u_i, v_i, and w_i, for $i = 2,\ldots,5$ are given below:

$$u_{m,2} = (b_m(2) + b_m(3)\frac{\Gamma_5+\Gamma_1\Gamma_6}{1-\Gamma_6} + b_m(4)\Gamma_6)/\Delta, \qquad u_{m,3} = \frac{b_m(3) + u_{m,2}(\Gamma_1 + \Gamma_5)}{1-\Gamma_6},$$

$$u_{m,4} = b_m(4) + u_{m,2}(\Gamma_2+\Gamma_6) + u_{m,3}\Gamma_1, \qquad u_{m,5} = b_m(5) + u_{m,2}\Gamma_3 + u_{m,3}\Gamma_2 + u_{m,4}\Gamma_1. \tag{5.25}$$

$$v_2 = (\Gamma_2 + \Gamma_3\frac{\Gamma_5+\Gamma_1\Gamma_6}{1-\Gamma_6} + \Gamma_4\Gamma_6)/\Delta, \qquad v_3 = \frac{\Gamma_3 + v_2(\Gamma_1 + \Gamma_5)}{1-\Gamma_6},$$

$$v_4 = \Gamma_4 + v_2(\Gamma_2+\Gamma_6) + v_3\Gamma_1, \qquad v_5 = \Gamma_5 + v_2\Gamma_3 + v_3\Gamma_2 + v_4\Gamma_1. \tag{5.26}$$

$$w_2 = (\Gamma_1+\Gamma_3+(\Gamma_2+\Gamma_4)\frac{\Gamma_5+\Gamma_1\Gamma_6}{1-\Gamma_6} + \Gamma_6(\Gamma_3+\Gamma_5))/\Delta, \qquad w_3 = \frac{\Gamma_2+\Gamma_4 + w_2(\Gamma_1+\Gamma_5)}{1-\Gamma_6},$$

$$w_4 = \Gamma_3+\Gamma_5 + w_2(\Gamma_2+\Gamma_6) + w_3\Gamma_1, \qquad w_5 = \Gamma_4+\Gamma_6 + w_2\Gamma_3 + w_3\Gamma_2 + w_4\Gamma_1. \tag{5.27}$$

5.2. Computational Experience

Both Algorithm 1 (the iterative algorithm) and Algorithm 2 performed strikingly similar to the corresponding algorithms for the exhaustive service case. We do not report on their performance here. To get some feel for numerical values, we revisit some of the examples given in section 4, and present the results obtained. We present the results for Examples 3 through 6 of section 4. These are the examples with a relatively large number of nodes, and high utilizations. For the gated service system, we can again make use of the "Computational Remark" in section 3. For all cases reported, we ran Algorithm 2 at level 6.

Example 3 revisited. Both algorithms obtained the following mean waiting times: $W_1 = 58.9669$, $W_2 = 46.2956$, $W_3 = 46.2918$, $W_4 = 46.2874$, $W_5 = 46.2822$, $W_6 = 45.4192$, $W_7 = 45.4182$, $W_8 = 45.4171$, $W_9 = 44.5587$, $W_{10} = 44.5788$.

Example 4 revisited. The mean waiting times ranged from 916.6964 at node 1 to 1061.0240 at node 11. Algorithm 2 obtained the *exact* mean waiting times (correct to 4 decimal places). In fact, it obtained the exact mean waiting times even at level 5.

Example 5 revisited. The mean waiting times ranged from 235.8342 at node 24 to 309.7431 at node 1. Algorithm 2 obtained the *exact* mean waiting times (correct to 4 decimal places).

Example 6 revisited. The mean waiting times ranged from 291.2991 at node 48, to 310.0504 at node 2. Algorithm 2 obtained the *exact* mean waiting times (correct to 4 decimal places) in 861.59 milliseconds.

6. Polling systems with a mix of gated and exhaustive service disciplines

In this section, we present the iterative algorithm for a polling system in which some of the nodes may be serviced according to the exhaustive service discipline, while other nodes may be served using the gated service discipline. The development of an algorithm similar to Algorithm 2, for this case, is left as a topic for possible future research.

The analysis, again, proceeds in a manner similar to the analysis for either the pure exhaustive service polling system, or for the pure gated service polling system. Let G denote the set of nodes using the gated service discipline, and E denote the set of nodes using the exhaustive service discipline. As before, we obtain the mean waiting times at the nodes, after computing the first and second factorial moments of the number of customers present at the node when the server polls it. The mean waiting time at a node, m, is given in terms of the f_m and $f_m^{(2)}$ as:

$$W_m = \frac{f_m^{(2)}}{2f_m}\,\frac{1+\rho_m}{\lambda_m}, \quad m \in G, \qquad \text{and} \qquad W_m = \frac{f_m^{(2)}}{2\lambda_m f_m} + \frac{\lambda_m b_m^{(2)}}{2(1-\rho_m)}, \quad m \in E. \tag{6.1}$$

To compute f_1 and $f_1^{(2)}$, we redefine the recursive funtion γ as follows:

$$\gamma_m(0) = z_m,\ m = 1,\ldots,M, \tag{6.2a}$$

$$\gamma_m(j) = \beta_m\Big(\sum_{k\le m}[\lambda_k-\lambda_k\gamma_k(j-1)] + \sum_{k>m}[\lambda_k-\lambda_k\gamma_k(j)]\Big), \quad j>0,\ m\in G, \tag{6.2b}$$

$$\gamma_m(j) = \eta_m\Big(\sum_{k<m}[\lambda_k-\lambda_k\gamma_k(j-1)] + \sum_{k>m}[\lambda_k-\lambda_k\gamma_k(j)]\Big), \quad j>0,\ m\in E. \tag{6.2c}$$

The term $\sigma_m(j)$ is still defined by equation (2.5), with the γ_m terms as defined above. Proceeding exactly as before, we obtain:

$$F_1(\gamma_1(0),\ldots,\gamma_M(0)) = \prod_{j=0}^{\infty}\Big(\prod_m \sigma_m(j)\Big). \tag{6.3}$$

Using the same definitions as before, equations (2.8), (2.9), and (2.12) still hold. Equation (2.11) is modified as follows. For $j > 0$,

$$\psi_m(j) = \rho_m\Big[\sum_{k\le m}\psi_k(j-1) + \sum_{k>m}\psi_k(j)\Big], \quad m\in G, \tag{6.4a}$$

$$\psi_m(j) = \delta_m\Big[\sum_{k<m}\psi_k(j-1) + \sum_{k>m}\psi_k(j)\Big], \quad m\in E, \tag{6.4b}$$

$$\psi_m^{(2)}(j) = \rho_m\Big[\sum_{k\le m}\psi_k^{(2)}(j-1)+\sum_{k>m}\psi_k^{(2)}(j)\Big] + \lambda_m b_m^{(2)}\Big[\sum_{k\le m}\psi_k(j-1)+\sum_{k>m}\psi_k(j)\Big]^2, \quad m\in G, \tag{6.4c}$$

$$\psi^{(2)}_m(j) = \delta_m \Big[\sum_{k<m} \psi^{(2)}_k(j-1) + \sum_{k>m} \psi^{(2)}_k(j) \Big] + \frac{\lambda_m b^{(2)}_m}{(1-\rho_m)^3} \Big[\sum_{k<m} \psi_k(j-1) + \sum_{k>m} \psi_k(j) \Big]^2, \quad m \in E, \qquad (6.4d)$$

Following a similar analysis as used to obtain f_1 for either the pure exhaustive service case, or the pure gated service case, we get

$$f_1 \;=\; \lambda_1 \frac{s}{1-\rho}, \text{ if node } 1 \in G, \qquad \text{and} \qquad f_1 \;=\; \lambda_1 s \frac{1-\rho_1}{1-\rho}, \text{ if node } 1 \in E, \qquad (6.5)$$

The term $f^{(2)}_1$ has an identical expression as the one for either the pure exhaustive or the pure gated case:

$$f^{(2)}_1 \;=\; f^2_1 + \sum_m s_m \sum_k y_k + \sum_m \mathrm{Var}(S_m) \sum_{j=0}^{\infty} \Big[\sum_{k\le m} \psi_k(j) + \sum_{k>m} \psi_k(j+1) \Big]^2. \qquad (6.6)$$

where

$$\sum_m y_m \;=\; \sum_m \frac{\lambda_m b^{(2)}_m}{(1-\rho)\rho^2_m} \sum_{j=1}^{\infty} \psi^2_m(j). \qquad (6.7)$$

Unlike either the pure exhaustive service case, or the pure gated service case, we must now take care when substituting for the term $[\sum_{k\le m} \psi_k(j) + \sum_{k>m} \psi_k(j+1)]^2$. For the pure gated service case, we used equation (5.4a) to write this term as $\psi_m(j+1)/\rho^2_m$, and we can still do that here. For the pure exhaustive service case, we used equations (2.20) and (2.21) to set this term equal to $\psi_{m+1}(j+1)/\rho^2_{m+1}$, for $j > 0$. However, this assumes that node m+1 also uses the exhaustive service discipline, which may no longer be true for the system we are considering. Nevertheless, we can still simplify this term for $m \in E$, as $[\sum_{k\le m} \psi_k(j) + \sum_{k>m} \psi_k(j+1)]^2 = [\psi_m(j) + \psi_m(j+1)/\delta_m]^2$, where $\delta_m = \rho_m/(1-\rho_m)$. Therefore, if we let

$$\chi_m(j) \;=\; \frac{\psi_m(j)}{\lambda_1}, \qquad (6.8)$$

$$\varphi_m(0) \;=\; \sum_{j=1}^{\infty} \chi^2_m(j), \qquad \text{and} \qquad \xi_m(0) \;=\; \sum_{j=0}^{\infty} [\chi_m(j) + \chi_m(j+1)/\delta_m]^2, \qquad (6.9)$$

from equations (6.6) and (6.7) we obtain Lemma 6.1:

Lemma 6.1:

$$f^{(2)}_1 \;=\; f^2_1 + \lambda^2_1 s \sum_m \frac{\lambda_m b^{(2)}_m}{(1-\rho)\rho^2_m} \varphi_m(0) + \lambda^2_1 \sum_{m\in G} \frac{\mathrm{Var}(S_m)}{\rho^2_m} \varphi_m(0) \;+\; \lambda^2_1 \sum_{m\in E} \mathrm{Var}(S_m)\, \xi_m(0). \qquad ■$$

From Lemma 6.1 and equations (6.1) and (6.5), we obtain:

Theorem 6.2: The mean waiting time at node 1 for the polling system with the mixed service discipline is:

$$W_1 = \frac{s}{2}\frac{1+\rho_1}{1-\rho} + (1+\rho_1)\sum_m \frac{\varphi_m(0)}{\rho_m^2}\frac{\lambda_m b_m^{(2)}}{2} + (1+\rho_1)\frac{1-\rho}{2s}\left(\sum_{m\in G}\frac{\mathrm{Var}(S_m)\varphi_m(0)}{\rho_m^2} + \sum_{m\in E}\mathrm{Var}(S_m)\xi_m(0)\right),$$

if node $1 \in G$, (6.10a)

$$W_1 = \frac{s}{2}\frac{1-\rho_1}{1-\rho} + \sum_m \frac{\varphi_m(0)}{(1-\rho_1)\rho_m^2}\frac{\lambda_m b_m^{(2)}}{2} + \frac{1-\rho}{2s(1-\rho_1)}\left(\sum_{m\in G}\frac{\mathrm{Var}(S_m)\varphi_m(0)}{\rho_m^2} + \sum_{m\in E}\mathrm{Var}(S_m)\xi_m(0)\right)$$

$$+ \frac{\lambda_1 b_1^{(2)}}{2(1-\rho_1)},$$

if node $1 \in E$, (6.10b)

■

6.1 Computational Experience:

We report on some of the numerical results for the mixed service discipline system. We use the data presented in examples 2 through 4, from section 3.

Example 1 revisited. The same data applies as before except that nodes 1, 3, and 4 use the gated service discipline, while nodes 2 and 5 use the exhaustive service discipline. The mean waiting times are W_1 = 139.5932, W_2 = 76.1434, W_3 = 147.2045, W_4 = 152.6066, W_5 = 111.6857.

Example 2 revisited. Nodes 1, 3, and 6 use the gated service discipline, while the other nodes use the exhaustive service discipline. The mean waiting times are W_1 = 340.1163, W_2 = 216.5708, W_3 = 358.8866, W_4 = 247.5317, W_5 = 290.8190, W_6 = 327.9425, W_7 = 275.1886.

Example 3 revisited. Nodes 1, 2, 3, 6, and 9 use the gated service discipline, while the other nodes use the exhaustive service discipline. The mean waiting times are W_1 = 59.3568, W_2 = 46.6172, W_3 = 46.6183, W_4 = 39.7035, W_5 = 39.6888, W_6 = 45.7251, W_7 = 40.5487, W_8 = 40.5423, W_9 = 44.8510, W_{10} = 41.4468.

7. Summary and Conclusions

We have presented a powerful technique, which we term the Individual Station (IS) technique for obtaining the mean waiting time at one or more nodes in a continuous-time polling system which uses either the pure exhaustive service discipline, the pure gated service discipline, or a system with a mix of both disciplines. The primary advantage of the IS technique is that one can now obtain the mean waiting time for a select subset of the nodes in the polling system. The techniques developed in the past require the simultaneous computation of the mean waiting times at all nodes. We developed two efficient algorithms, one based on an iterative approach (Algorithm

1), and the other based on solving a small system of equations of size 6 or less (Algorithm 2), to obtain the mean waiting times. Algorithm 2 requires $O(M^2)$ operations to find the mean waiting time at a node. In this paper, Algorithm 2 was developed only for the pure exhaustive, and the pure gated service systems.

We presented a number of compuational results for the pure exhaustive service polling system, the pure gated service polling system, and the mixed service polling system. The computational results suggest that the iterative algorithm requires roughly the same number of iterations for a given value of ρ, for any number of nodes. This also appears to suggest that the iterative algorithm would probably perform better than an $O(M^3)$ algorithm in computing the mean waiting times at all nodes, so long as ρ is less than 0.90.

Algorithm 2 does not depend on the value of ρ for its execution time. This algorithm is exact for the exhaustive service (gated service) system with 7 (6) nodes. However, it performs remarkably accurately in computing the mean waiting times. We experimented with ρ values very close to 1, and found that the maximum percentage error from the exact mean waiting time was less than 0.0025%. (For all practical purposes, this would characterize the algorithm as an exact algorithm.) These errors could be reduced even further, if needed, with little increase in computational effort, simply by executing the algorithm at a higher level (i.e., for the exhaustive service system, this would involve solving a system of 7 equations, instead of a system of six equations).

The IS technique would be especially useful when evaluating several alternative system configurations in the preliminary design stages of, say, a computer network, where the analyst is usually interested in obtaining very accurate measures of performance only for a select few nodes in the network.

There are a number of topics which remain to be explored, on applications of the IS technique. It would be of interest to consider applying the IS technique to other polling systems such as, for example, the nondeterministic polling system (Srinivasan 1991a, Boxma and Weststrate 1989). We have recently developed an iterative algorithm (Srinivasan 1991b) to determine waiting time variances for the polling systems considered in this paper.[1]

[1] The author thanks K.G. Murty and R. Saigal for their helpful comments.

References

Y. Aminetzah. 1975. An exact approach to the polling system. Ph.D. dissertation, McGill University, Montreal, P.Q., Canada.

J.E. Baker and I. Rubin. 1987. Polling with a general-service order table. *IEEE Trans. on Communications*, v 35, pp. 283-288.

O.J. Boxma and J.A. Weststrate. 1989. Waiting times in polling systems with Markovian server routing. *Proc. Conference Braunschweig*, Springer Verlag.

R.T. Carsten, E.E. Newhall, and M.J.M. Posner. 1977. A simplified analysis of scan times in an asymmetrical Newhall loop with exhaustive service. *IEEE Transactions on Communications*, v 25, pp. 951-957.

R.B. Cooper. 1970. Queues served in cyclic order: waiting times. *Bell System Tech. Jnl.,* v 49, pp. 399-413.

R.B. Cooper and G. Murray. 1969. Queues served in cyclic order. *Bell System Tech. Jnl.*, v 48, pp. 675-689.

R.B. Cooper. 1981. *Introduction to queueing theory*, Second Edition, Elsevier Science Publishers, New York.

M. Eisenberg. 1972. Queues with periodic service and changeover time. *Opns. Research*, v 20, pp. 440-451.

M.J. Ferguson and Y.J. Aminetzah. 1985. Exact results for nonsymmetric token ring systems. *IEEE Transactions on Communications*, v 33, pp. 223-231.

M. Fiedler, and V. Ptak. 1962. On matrices with nonpositive off diagonal elements and positive principal minors. *Czechoslovakian Mathematical Journal*, v 12, 1962, pp. 382-400.

O. Hashida. 1972. Analysis of multiqueue. *Rev. Elect. Commns. Lab.*, v 20, pp. 189-199.

P. Humblet. 1978. Source coding for communciation concentrators. ESL-R-798, Electron Syst. Lab., Massachussets Institute of Technology, Cambridge, MA.

L. Kleinrock and H. Levy. 1988. The analysis of random polling systems. *Opns. Res.*, v 36, pp. 716-732.

A.G. Konheim and B. Meister. 1974. Waiting times and lines in systems with polling. *Jnl. Association for Computing Machinery*, v 21, pp. 470-490.

H. Levy. 1989. Delay computation and dynamic behavior of non-symmetric polling systems. *Performance Evaluation*, v 10, pp. 35-51.

H. Levy. 1991. A note on the complexity of Swartz's method for calculating the expected delay in non-symmetric cyclic polling systems. *Operations Research Letters*, v 10, pp. 363-368.

I. Rubin and L.F. DeMoraes. 1983. Message delay-analysis for polling and token multiple-access schemes for local communication networks. *IEEE Transactions on Selected Areas in Communications*, v 1, pp. 935-947.

G.B. Swartz. 1980. Polling in a loop system. *J. Assoc. for Comp. Machinery*, v 27, pp. 42-59.

D. Sarkar and W.I. Zangwill. 1989. Expected waiting time for nonsymmetric cyclic queueing systems - exact results and applications. *Management Science*, v 35, pp. 1463-1474.

M.M. Srinivasan. 1991a. Non-deterministic polling systems. *Management Science*, v 37, no 6, pp. 667-691.

M.M. Srinivasan. 1991b. Waiting time variances in cyclic service systems. Technical Report 91-37, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109-2117, December 1991.

H. Takagi. 1986. *Analysis of polling systems*, MIT Press, Cambridge, MA.

H. Takagi. 1990. Queueing analysis of polling models: an update. *Stochastic Analysis of Computer and Commn. Systems*, Elsevier Science Publishers B.V. (North-Holland), Amsterdam.